

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки
(повне найменування інституту, факультету)

Кафедра обчислювальної техніки
(повна назва кафедри)

До захисту допущено:

Завідувач кафедри

_____ **Сергій СТИРЕНКО**
(підпис) (ім'я, прізвище)

« ____ » _____ 2020 р.

Дипломний проект

на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп’ютерні системи
та мережі” спеціальності 123 “Комп’ютерна інженерія”

на тему “Система розумного дому з голосовим управлінням”

Виконав: студент _____ **IV** _____ курсу, групи **ІО-64**

_____ **Осадчий Юрій Олександрович**

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник _____ **доцент Русанова Ольга Веніамінівна**

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Консультант _____ **н. контроль** _____ **доц. д. т. н. Сімоненко В. П.**

(назва розділу)

(вчені ступінь та звання, прізвище, ініціали)

_____ (підпис)

Рецензент доц. каф. _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ - 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки
(повне найменування інституту, факультету)

Кафедра обчислювальної техніки
(повна назва кафедри)

Рівень вищої – освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія» Освітньо-

професійна програма – «Комп'ютерні системи та мережі»

До захисту допущено:

Завідувач кафедри

_____ **Сергій СТИРЕНКО**
(підпис) (ім'я, прізвище)

« ____ » _____ 2020 р.

ЗАВДАННЯ

на дипломний проект студенту

_____ **Осадчому Юрію Олександровичу**

1. Тема проекту **“Система розумного дому з голосовим управлінням”**, керівник проекту _____ **Русанова**
Ольга Веніамінівна доцент , затверджені наказом по університету від «
07 » травня 2020р. № 1081-с _____
2. Термін подання студентом проекту _____ **2020р.**
3. Вихідні дані до проекту: **технічне завдання, теоретичні дані, наукові публікації.**
4. Зміст пояснювальної записки: **огляд доступних систем розумного дому, поставлення задачі для реалізації системи розумного дому з голосовим управлінням, проектування та розробка програмного та технічного продукту, моделювання та аналіз отриманих результатів.**
5. Консультант роботи, з вказівкою розділів роботи, які до них вносяться

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко В.П.		

6. Дата видачі завдання 01.09.2019 року

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проекту (роботи)	Строк виконання етапів проекту(роботи)	Примітки
1	Затвердження теми роботи	01.09.2019	
2	Вивчення та аналіз завдання	14.09.2019	
3	Розробка архітектури та загальної структури систем	12.10.2019	
4	Розробка структур окремих підсистем	26.10.2019	
5	Програмна реалізація системи	30.11.2020	
6	Оформлення пояснювальної записки	28.03.2020	
7	Передзахист	26.05.2020	
8	Захист	25.06.2020	

Студентка Юрій ОСАДЧИЙ

(підпис)

Керівник Ольга РУСАНОВА

(підпис)

Анотація

Бакалаврська дипломна робота присвячена вирішенню проблеми доступності систем розумного дому з голосовим управлінням для людей з обмеженими можливостями, для людей похилого віку та широкого загалу користувачів. Розглянута модель вирішує проблеми зі зручністю встановлення, налаштування та користування данною системою. Розроблений програмний та технічний продукт забезпечує можливість створення системи розумного дому з голосовим управлінням для недосвічених користувачів.

Annotation

The bachelor's thesis is devoted to solving the problem of accessibility of smart home systems with voice control for people with disabilities, for the elderly and the general public. This model solves problems with the ease of installation, configuration and use of this system. Developed software and hardware product provides the ability to create a smart home system with voice control for untrained users.

ТЕХНІЧНЕ ЗАВДАННЯ

до дипломної роботи
освітньо-кваліфікаційного рівня бакалавр

на тему: «Система розумного дому з голосовим управлінням»

Київ – 2020 року

ЗМІСТ

1.	Найменування та область застосування	2
2.	Підстава для розробки	2
3.	Призначення розробки	2
4.	Джерела розробки	2
5.	Технічні вимоги	2
5.1	Вимоги до розроблюваного продукту	2
5.2	Вимоги апаратного забезпечення	3
5.3	Вимоги до надійності.....	3
6.	Етапи розробки	3

					ІАЛЦ. 466454.002.ТЗ			
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив	Осадчий Ю.О.				Система розумного дому з голосовим управлінням	Лит.	Лист.	Аркушів
Перевірів	Русанова О.В						1	3
Реценз.						НТУУ "КПІ", ФІОТ, ІО-64		
Н. Контр.	Сімоненко В. П.							
Затвердив					Технічне завдання			

1. Найменування та область застосування

Система розумного дому з голосовим управлінням. Область застосування — домашнє використання.

2. Підстава для розробки

Підставою для розробки є індивідуальне завдання на реалізацію системи розумного дому з голосовим управлінням, затверджене кафедрою Обчислювальної техніки ФІОТ НТУУ “КПІ ім. Ігоря Сікорського”.

3. Призначення розробки

Даний проект призначений для моделювання системи розумного дому з голосовим управлінням, а також для демонстрації набутих навичок та закріплення отриманих знань.

4. Джерела розробки

Основними джерелами розробки є науково-технічна література по програмному забезпеченню, наукові публікації щодо систем на базі Arduino та дослідження в області розпізнавання мови та перекладу мовлення в текст.

5. Технічні вимоги

5.1 Вимоги до розроблюваного продукту

Розробка інтерфейсу для моделювання системи розумного дому з голосовим управлінням.

Збереження ключових слів в базі даних, реалізація доступу до даних системі розумного дому.

					ІАЛЦ. 466454.002.ТЗ			
Зм.	Арк.	№ докум	Підпис	дата	Система розумного дому з голосовим управлінням	Лит	Лист	Архів
Розробив	Осадчий Ю.О.						2	3
Перевір	Русанова О.В.							
Н. контр.	Сімоненко В.П.							
Затверд.					Технічне завдання	НТУУ “КПІ”, ФІОТ, ІО-64		

Демонстрація результатів роботи системи.

5.2 Вимоги програмного забезпечення

Ubuntu Linux 14.x+, Mac OS 10.8.3+, 10.9+, MS Windows 10.

Інтеграційне середовище розробки Android studio, Arduino IDE.

5.2 Вимоги апаратного забезпечення

Для Windows: RAM 512 МБ; мінімальна вимога до процесора
Pentium 2 266 МГц

5.3 Вимоги до надійності

Проходження тестування у процесі роботи системи

Строк використання — необмежений

6. Етапи розробки

1. Дослідженні існуючих рішень	01.09.2019
2. Постановка та аналіз технічного завдання	14.09.2019
3. Розробка архітектури та основних модулів системи	12.10.2019
4. Тестування окремих модулів та усієї системи	26.10.2019
5. Налаштування програмного продукту	30.11.2020
6. Написання документації до дипломної роботи	28.03.2020

					ІАЛЦ. 466454.002.ТЗ			
Зм.	Арк.	№ документа	Підпис	Дата	Система розумного дому з голосовим управлінням	Лит.	Лист.	Аркушів
Розробив		Осадчий Ю.О.						
Перевірив		Русанова О.В.					3	3
Реценз.						НТУУ "КПІ", ФІОТ, ІО-64		
Н. Контр.		Сімоненко В. П.						
Затвердив					Технічне завдання			

Пояснювальна записка
до дипломного проекту
на тему: «Система розумного дому з голосовим управлінням»

Київ – 2020 року

Зміст

ВСТУП.....	5
РОЗДІЛ 1. ПРЕДСТАВЛЕННЯ ГОТОВИХ РІШЕНЬ	6
1.1 Доступність смартфону та доступність персональних комп'ютерів.	6
1.2 Інтернет речей.	8
1.3 Домашня автоматизація.	8
1.4 Поточний стан домашньої автоматизації.	9
1.5 Популярність домашньої автоматизації.	9
1.6 Інтернет речей з домашньою автоматизацією та голосовими командами.	10
1.7 Теперішнє та майбутнє домашньої автоматизації.	10
1.8 Технологічні стандарти	11
1.9 Популярні платформи для домашньої автоматизації	12
1.9.1 Apple HomeKit.....	14
1.9.2 Термостат Google Nest.....	15
1.9.3 IFTTT (IF-THIS-THEN-THAT)	17
1.9.4 Amazon Echo.....	18
1.9.5 Belkin's Wemo Switch	19
Висновки до розділу 1.	21
РОЗДІЛ 2. ПРЕДСТАВЛЕННЯ СВОГО РІШЕННЯ	22
2.1 Вступ до домашньої автоматизації з можливістю управління голосом. ...	22
2.2 Причини використання Wi-Fi замість ZigBee та X10	22
2.3 Мікроконтролер Arduino.....	24

					ІАЛЦ.467100.003 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Система розумного дому з голосовим управлінням Пояснювальна записка	Літ.	Аркуш	Аркушів
Розробив		Осадчий Ю.О.						
Перевір.		Русанова О.В.					1	58
Н. контр.		Сімоненко В.П.				НТУУ "КПР", ФІОТ, ІО-64		
Затверд.								

2.4	Компоненти домашньої автоматизації.	26
2.5	Блок управління IVA.....	28
2.6	Підключення пульта управління:.....	28
2.7	Підключення та налаштування Wi-Fi.	30
2.8	Фізичні з'єднання	31
2.9	Налаштування параметрів.....	32
2.10	Отримання команди з програми.....	34
2.11	Структура команд.	34
2.12	Контроль фізичних пристроїв.	35
2.13	Фрагмент коду методу "налаштування" для панелі керування	36
	Висновки до розділу 2.	39
	Розділ 3. Програмне забезпечення та інтеграція IVA	40
3.1	Інтерфейс користувача.	40
3.2	XML код.....	40
3.3	Пояснення.	43
3.4	Кнопка перемикання.....	44
3.5	Активація мікрофона для голосових команд.	45
3.6	Конвертація мови в текст.	48
3.6.1	Обробка мовлення.	48
3.7	Код підключення та комунікації сокета.	49
3.8	Код фільтрації ключових слів.....	50
	Висновки до розділу 3.	52

Розділ 4	53
Висновки та розвиток системи в майбутньому.....	53
4.1 Висновки.....	53
4.2 Майбутній розвиток системи.....	53
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	56

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		3

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

IVA	(Intelligent Voice Activated) Інтелектуальна система розумного до дому з можливістю управління голосом.
HCI	(Human Computer Interface) Інтерфейс взаємодії з комп'ютером за допомогою зрозумілих людині засобів зв'язку.
IoT	(internet of things) Інтернет речей.
RFID	(Radio frequency identification) радіочастотна ідентифікація.
WWDC	(Apple Worldwide Developers Conference) Всесвітня конференція для розробників Apple.
IFTTT	(IF-THIS-THEN-THAT) Безкоштовна веб-служба, за допомогою якої можна створювати прості умовні ланцюжки команд для автоматизації.
IP	(Internet Protocol) Протокол мережевого рівня для передавання даних між мережами.
WLAN	(Wireless Local Area Network) Бездротова локальна мережа.
LAN	(Local Area Network) Локальна комп'ютерна мережа.
IDE	(Integrated development environment) Інтегроване середовище розробки.
GUI	(Graphical user interface) Графічний інтерфейс користувача.
SSID	(Service Set Identifier) Унікальне найменування бездротової мережі.

ВСТУП

Ця робота представляє собою макет оригінальної інтелектуальної системи розумного дому з голосовим управлінням. Моя робота була розділена на три частини. В першій частині розглядаються найбільш успішні та найбільш популярні комерційні архітектури систем розумного дому з голосовим управлінням та систем Home-Automation.

На цьому етапі розробки я намагаюсь отримати глибоку оцінку різноманіття організацій, можливостей, обмежень та потенціальних областей росту існуючих ведучих систем домашньої автоматизації. Щоб отримати знання в цій предметній області мені довелось використовувати зворотну інженерію. Причиною використання цієї методики є той факт, що всі комерційні продукти що оглядаються в цьому дослідженні, комерційно захищені як промислові секрети. Виходячи з цього, неможливо отримати детальний опис їх реальної внутрішньої структури, архітектури та внутрішніх операцій.

В другій та третій частині цієї роботи міститься мій особистий вклад у вигляді прототипу для архітектури розумного дому. Моя розробка має в своїй основі Intelligent Voice Activated(IVA) для домашньої автоматизації, головним чином побудована на обробці голосових команд на природній мові. Я стверджую, що такий підхід повинен бути привабливим як для похилих людей та людей з обмеженими руховими можливостями так і для широкого кола користувачів. Крім того обладнання, що необхідне для реалізації системи являється загальнодоступним та недорогим. Найскладнішим елементом в моїй системі являється смартфон, котрий в більшості випадків вже мають потенційні користувачі.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		5

РОЗДІЛ 1.

ПРЕДСТАВЛЕННЯ ГОТОВИХ РІШЕНЬ

1.1 Доступність смартфона та доступність персональних комп'ютерів.

Оскільки комп'ютери стають більш продуктивними, меншими за розміром та дешевшими, вони перехоплюють більше аспектів звичайного життя людини. Але сьогодні обчислювальна потужність смартфона порівняна з великим настільним комп'ютером. Портативні пристрої такі як смартфони та планшети, отримують ще більший досвід в розумінні навколишнього середовища за допомогою вбудованих датчиків та сенсорів. Як повідомляють різноманітні джерела AVC [1], Forbes [22], Business Inside [21] смартфони стали найбільш швидкозростаючою галуззю в 21 столітті. Діаграма, що показана знизу, відображає як ринок смартфонів обганяє традиційну долю ринку персональних комп'ютерів.

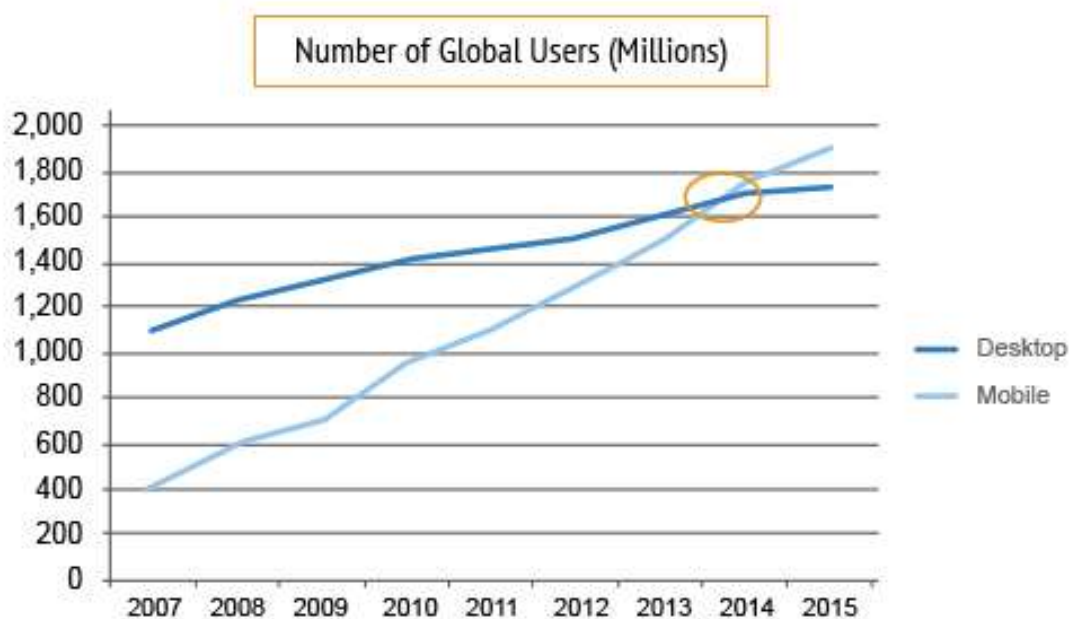


Рисунок 1.1 – Співвідношення мобільного трафіку та трафіку з ПК.

На рисунку 1.1 ми можемо побачити, що прийняття портативних комп'ютерів, тобто смартфонів та планшетів – це масове явище у всьому світі.

Очевидно, що вихід у широке розповсюдження портативних пристроїв полягає в тому, що суспільству стає більш комфортно з мобільними технологіями.

Якщо ми подивимось на десять чи двадцять років назад, то побачимо, що технологія тоді мала набагато меншу обчислювальну потужність та мала меншу інтеграцію з суспільством аніж сьогодні. Це був час персональних комп'ютерів.

Багато дослідів показують, що люди похилого віку почали набагато швидше приймати смартфони аніж декілька років тому. До того ж, я показують дослідження у всьому світі, люди похилого віку яку переважно старші за 65 років, мали труднощі у вивченні комп'ютерів раніше. Для них комп'ютери були громіздкими, мали меншу користь та вважалися надто складними для навчання та роботи. Але зараз люди похилого віку стають все більш відкритими для використання смартфонів. Це в основному результат того, що мобільний пристрій портативний. Корисний набір датчиків та сенсорів, що вбудовані в мобільний пристрій, таких як GPS, камера, моніторинг фізичного стану [4], а також розпізнавання голосу допомагають людям отримати більший досвід використання смартфона. До того ж мобільні пристрої забезпечують відчуття безперервного доступу до мережі інтернет. За допомогою цього можна отримати корисну інформацію та потрібні послуги на «кінчиках пальців».

Також важливо те, що елементи Human Computer Interface (HCI) стали простішими для використання природніх жестів та дотиків в порівнянні з комп'ютерною мишкою та клавіатурою. Отже, смартфони діють як посередники, приносячи інформацію та послуги для широкого кола користувачів роблячи це в більш інтуїтивний спосіб.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		7

1.2 Інтернет речей.

Оскільки комп'ютери стають все більш ефективними в обробці складних і великих даних з використанням хмарних обчислювальних технологій, це стає звичайною практикою щоб зробити окремі традиційні електронні прибори автоматизованими за допомогою окремого комп'ютера. Наприклад: телевізори, мікрохвильові печі, пральні машини, та навіть домашні пристрої для приготування кави стають більш «розумними». Побутова техніка нового покоління виготовляється з розрахунком на те, щоб зробити її тісно зв'язаною зі світом інтернету через мікроконтроллери. Наступною «великою річчю» в технології, як пророкують, будуть різні типи менших пристроїв, що взаємодіють як єдиний координований пристрій. Цей набір маленьких пристроїв, що працюють разом називається «інтернет речей» або IoT (internet of things). Це новий термін, що використовується для опису архітектурної моделі [11], в якій необмежена та зростаюча кількість приладів, що можуть бути підключені до інтернету.

1.3 Домашня автоматизація.

Одна із областей, що входять в основу для IoT – домашня автоматизація. Домашня автоматизація, також називається Dotomic Computing, дозволяє користувачу керувати побутовими пристроями за допомогою пульта дистанційного керування. Хоча домашня автоматизація не являється новою концепцією, ідея зробити домівку розумнішою завоювала популярність як побічний продукт успіху та розповсюдження концепції IoT. В теперішній час користувач може керувати побутовими пристроями за допомогою дотику до елемента керування в додатку чи певного жесту на своїх смартфонах чи планшетах.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		8

1.4 Поточний стан домашньої автоматизації.

Інтелектуальне програмне забезпечення, що вбудоване в потужні мікроконтролери та процесори, зробило більш доступним дешеве проектування та розробку пристроїв автоматизації. Таким чином домашня автоматизація стає дуже популярною. Компанії розробники програмного забезпечення, такі як Google, Amazon, Apple, Comcast, AT&T та інші або вже вступили в бізнес домашньої автоматизації, або планують зробити це найближчим часом [2]. Ми розглянемо продукт кожної компанії більш детально далі. Все це показує, що домашня автоматизація розглядається, як одне із наступних досягнень в технологіях.

1.5 Популярність домашньої автоматизації.

Керування домівкою не являється концепцією майбутнього. Керування приладами будинку за допомогою пульта теж не новина. Ми маємо телевізори, кондиціонери, музичні системи та навіть гаражні ворота, котрі керуються за допомогою спеціального пульта. Більшість цих фізичних приладів тепер реалізовані у вигляді програмних додатків, що працюють на смартфонах, планшетах та персональних комп'ютерах. До цього моменту ідея управління будинком за допомогою блоку керування з кнопками була широкого прийнята в світі. Настав день, щоб зробити домівки розумними, піддатливими, швидкими та швидко реагуючими, з меншою кількістю похибок та збоїв. До цього готові і технології і користувачі. Будинки сьогодні готові до керування з одного інтелектуального пульта. Вони стають достатньо розумними щоб автоматично вирішувати як прилади будуть взаємодіяти чи як адаптуватися до вимог сучасної людини. В сучасних будинках все встановлені інтелектуальні термостати та датчики, які можуть «спостерігати» та вивчати потреби своїх користувачів. Потім ця система може адаптуватися до нової звички користувача.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		9

Ми вже можемо бачити прості та дешеві продукти автоматизації навколо нас: відео-домофони, автоматичні дверні замки, RFID, датчики руху, інтелектуальні термостати та регулятори температури. Ринок розумного дому повний потреб та нових перспективних ідей. Такі рішення, як голосовий ввід на природній мові в розумному домі користуються великим попитом.

1.6 Інтернет речей з домашньою автоматизацією та голосовими командами.

Уявіть собі що розумний дім розмовляє з користувачем коли потрібно озвучити якусь інформацію. Дім запам'ятовує щоденні потреби користувача та піклується про них до того, як користувач дасть команду. Наприклад, якщо розумний дім нагадує користувачу взяти його офісні папки до того як піти на роботу. Або інший приклад, розумний дім може відкрити двері гаражу, якщо користувач виходить через парадні двері. Також, можливий приклад роботи, коли розумний дім робить ранкову каву як тільки користувач прокинувся від будильника. Все це розумний дім може зробити навіть не питаючи користувача. Це може бути досягнуте за допомогою інтеграції IoT в систему домашньої автоматизації. IoT може підключати такі пристрої, як пральні машини, холодильники, телевізори, кондиціонери, обігрівачі, датчики руху, ворота гаражу та багато інших типів пристроїв, разом з цим система може давати команди для виконання задач.

1.7 Теперішнє та майбутнє домашньої автоматизації.

Базові системи домашньої автоматизації сьогодні стали невід'ємною частиною життя людей. Такі системи як: датчики руху, інтелектуальне управління термостатом, світлові прибори, що можна налаштовувати, RFID, біометричні датчики, інфрачервоні технології, радіочастотні технології, системи відео нагляду та безпеки стали дуже поширеним явищем. Також передові технології, такі як

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		10

розпізнавання обличчя, голосові команди, допомога в плануванні повсякденного життя вже реалізовані. Популярність таких технологій швидко зростає. Сьогодні на ринку представлено багато продуктів для домашньої автоматизації з різних цінових сегментів. У кожного з них є свій спосіб зв'язку з контроллером пристрою. Наприклад радіочастотні пульти вмикати або вимикати світло, розумну розетку, якою можна керувати через смартфон або планшет. Також дуже популярною є технологія розпізнавання руху. Відкриття дверей, як тільки хтось підходить є дуже хорошим прикладом автоматизації. Це всього декілька розповсюджених прикладів із великої кількості різноманітних, а головне інтелектуальних систем.

Більшість ведучих світових технологічних компаній працюють над створенням своїх власних версії технології домашньої автоматизації. Мі постараємось зрозуміти технології деяких продуктів, що вже представлені на ринку. Кожна компанія має свої унікальні проекти та реалізації. Також компанії, що розробляють такі технології співпрацюють з виробниками побутової техніки та електроприладів, щоб сторонні виробники мали змогу розробляти прилади, що мають сумісність з їх платформою. Наприклад, лампочка відтінку Phillips, яка змінює свій колір, тепер можна керувати за допомогою Apple. Домашній комплект і команди Siri. Для цього Phillips довелося співпрацювати з Apple. Платформа Apple спілкуватися зі своїми пристроями.

1.8 Технологічні стандарти

У цьому розділі обговорюються стандарти та протоколи, що найчастіше використовуються. Хоча більшість компаній реалізують власну версію стандартів, важко проаналізувати унікальну платформу кожного продукту. Таким чином, ми розглянемо порівняння з відкритим кодом та загально визнані стандарти.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		11

Сьогодні існує безліч стандартів та протоколів, впроваджених галузями. Кожна компанія намагається впровадити на ринок власну технологію. Інтернет Речей, або IoT - це комунікаційна платформа для легкої взаємодії мікроконтролера та пристроїв. У кожної компанії та організації є свій унікальний спосіб впровадження IoT. У деяких є свої маршрутизатори на основі ZigBee, необхідні для спілкування з ними власні виробничі пристрої, тоді як деякі компанії використовують Wi-Fi в якості базового бездротового зв'язку. Мережа. Bluetooth також широко використовується для проектування мереж на базі IoT. Деякі приклади системи домашньої автоматизації: домашній набір Apple, Samsung. Платформа SmartThings та рішення щодо автоматизації дому AT & T. Comcast теж має свою систему домашньої автоматизації. У кожного є свої пристрої, що спілкуються між собою. Таким чином, стандартизація є дуже фрагментарною в поточний час, яка включає Wi-Fi, ZigBee, Bluetooth, інфрачервоний або інші бездротові методи для формування виділеної невеликої мережі інтернету речей.

1.9 Популярні платформи для домашньої автоматизації

Для впровадження домашньої автоматизації необхідно три речі:

- Пристрій отримує введення користувачем для управління всіма діями, такими як дистанційне керування чи голосове апаратне забезпечення на основі розпізнавання.

Таблиця 1.1 - Порівняння мережевих технологій Wi-Fi та ZigBee

	Wi-Fi	ZigBee
Основні технології	Використовує роутер довшого діапазону для надсилання даних	Він призначений для перевезення малих кількостей даних за короткий час та відстань

	Wi-Fi	ZigBee
Тип мережі	Централізована мережа конденсаторного типу	Це сітчастий стандарт мережі, в якій всі вузли у мережі з'єднані між собою
Потужність споживання	Потрібно багато енергії, як порівняно із ZigBee	Дуже низьке енергоспоживання. Акумулятор може тримати заряд місяцями
Швидкість	до 5 Гбіт/с	250 Кбіт/с
Вартість	вище, ніж ZigBee, але доступні від 50 до 500 доларів	дуже дешево, 10 - 50 доларів
Тип інформації, що можна надсилати	Потокова музика, відео та відеодані з високою пропускнуою здатністю	Максимум можуть бути надіслані текстові дані.
Простота використання та реалізації	Багато інтернет-документації, та підтримка широкої громадськості, що використовувала та впроваджувала Wi-Fi мережу.	Дуже технічна для загальних користувачів. Складна для загального користування. Щоб зрозуміти та реалізувати технологію потрібні деякі знання щодо вбудованих систем.

1.9.1 Apple HomeKit

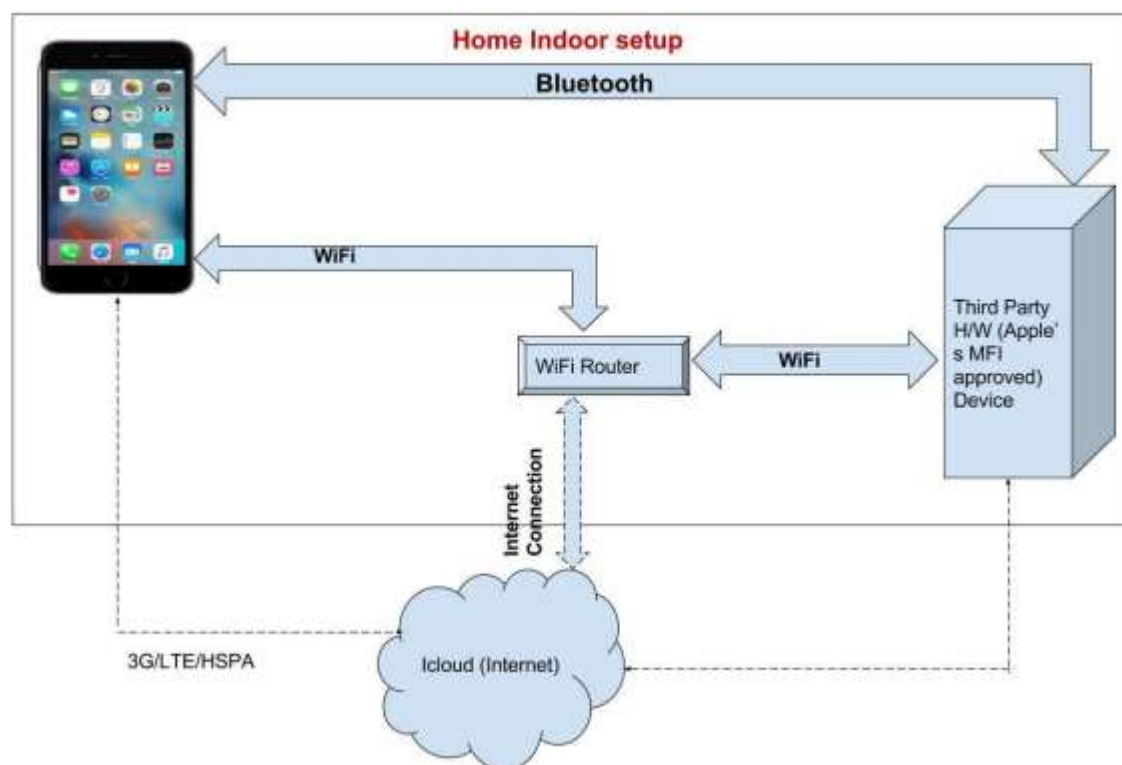


Рис. 1.2. Структура системи автоматизації Apple HomeKit.

У червні 2014 року на своїй Всесвітній конференції розробників (WWDC) Apple представила iOS 8 та виділила деякі з майбутніх функцій. Серед цих особливостей були дві нові Платформи “Kit”, HealthKit та HomeKit, розроблені для інтеграції iOS з двома тенденціями цифрового віку що зростають: кількісна оцінка самостійної охорони здоров'я та розумні будинки. HealthKit - це ваше обладнання для тіла та фітнесу, HomeKit - це ваш дім та електроніка. HomeKit не є єдиним контрольним додатком. Це платформа сертифікації обладнання, і система баз даних, яка дозволяє розробникам створювати обладнання. Більше того, це допомагає інтегрувати апаратне забезпечення з iOS для легкого відкриття, конфігурації, управління, та спілкування між найрізноманітнішими продуктами розумного дому, такими як замки, світильники, охоронне обладнання та інші

продукти домашньої автоматизації. Озброївшись продуктами HomeKit та пристроєм iOS, можна вбудувати їх в будинок таким чином, щоб вогні вмикались в певний час, щоб розбудити користувача. Крім того, наприкінці дня можна притулитися до ліжка і вголос поговорити зі своїм пристроєм iOS і сказати «HomeKit закрий будинок на ніч».

1.9.2 Термостат Google Nest

Термостат Nest Learning виходить за рамки простого регулювання температури для того, впливати на споживання енергії для споживачів. Щоб зрозуміти цінність Nest, давайте спочатку розглянемо, що роблять інші термостати. Всі термостати дозволяють вам встановлюєте бажану температуру та стежити за поточною температурою. Багато термостатів повністю покладаються на користувача, щоб встановити температуру. Хоча в останні роки, виробники запропонували програмовані термостати, які можуть допомогти заощадити енергію. Це допомагає користувачеві програмувати певні температури на певний час доби. Наприклад, дозволяючи їм автоматично знижувати температуру, коли вони будуть поза домом (щоб заощаджувати енергію). Однак через складність цих термостатів люди не завжди налаштовують їх правильно. Це може заперечувати більшість, якщо не весь, їх енергозберігаючий потенціал.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		15

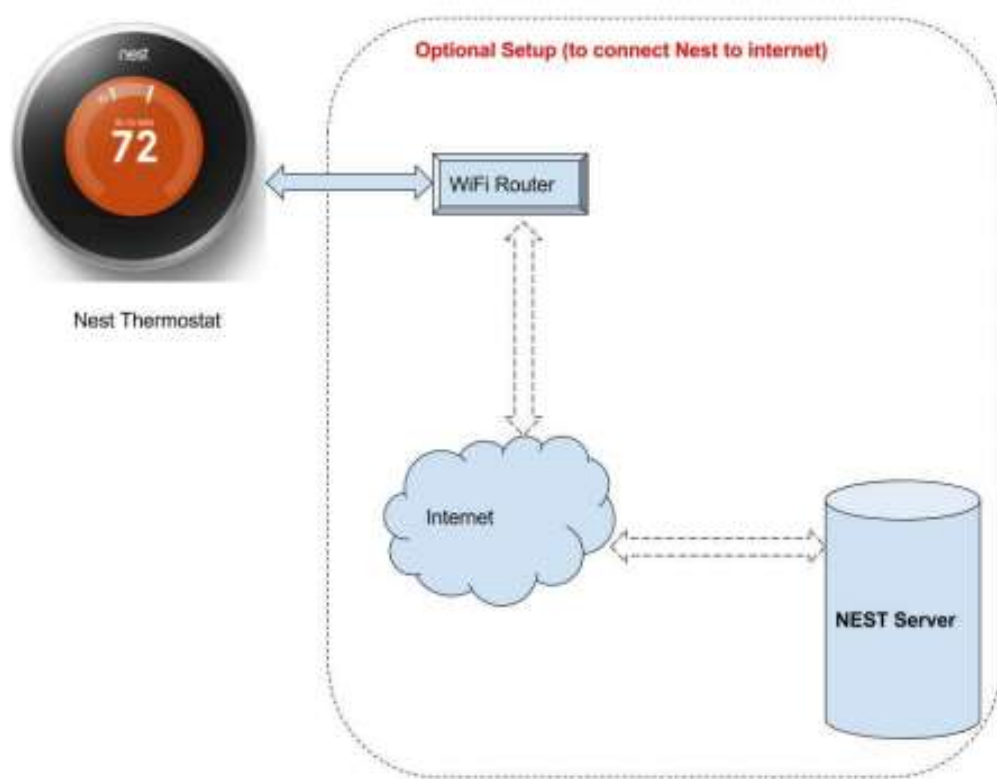


Рис. 1.3. Структура системи Google Nest.

- Nest використовує різні входи для дотримання щоденного розпорядку свого користувача та використовує їх для підтримки графіку автоматично, виходячи з того, що він вивчає. (Компанія називає цю технологію Nest Sense.)
- Nest створює режим автоматичного відключення на основі інформації про користувача.
- Встановлювати температуру також можна вручну, якщо бажає користувач.
- Поки термостат активно нагрівається або охолоджується, Nest відображає орієнтовний час роботи системи для досягнення бажаної температури.
- Nest відображає зелене листя будь-коли, коли терморегулятор працює з енергозбереженням

- Може допомогти навчити користувачів приймати енергозберігаючі рішення.
- Nest дозволяє користувачеві знати, яка діяльність (між автоматичним відключенням, власними налаштуваннями користувача) призводить до найбільшої економії енергії протягом дня.
- Nest використовує Wi-Fi для підключення до облікового запису користувача на nest.com. Ця функція дозволяє користувачам слідкувати та налаштовувати Nest віддалено з веб-сайту.
- Nest підтримує мобільний додаток, доступний для Apple iOS та пристроїв Android. Додаток перетворює мобільний пристрій у віддалений контролер за Nest..
- Обліковий запис Nest може керувати до 10 пристроями Nest.

1.9.3 IFTTT (IF-THIS-THEN-THAT)

IFTTT має веб-сайт, який спонукає користувачів створити свій власний тригер дії та умови, щоб змусити сумісні пристрої виконувати завдання. Дзвінки IFTTT сценарії з рецептами тригерів. Наприклад, якщо користувач просто робить сценарій використання, що датчик всередині його домашнього бару виявляє, що пляшка витягнута, тоді система може надсилати повідомлення(щоб слідкувати за дітьми поки ви не вдома). Якщо у вас є, наприклад, система розумного освітлення Philips Hue, користувач може користуватися нею за допомогою IFTTT. У ще одному прикладі користувач може використовувати IFTTT для автоматичного читання електронної пошти коли хтось коментує ваш блог WordPress. Існують численні комбінації(вони мають назву рецепти) на IFTTT, які можуть полегшити життя. IFTTT не є платформою домашньої автоматизації; це більш загальна платформа для виконання завдання, якщо спрацює тригер. Система може просто надіслати повідомлення на смартфон, якщо погода стає поганою. У цьому

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		17

прикладі не використовується домашня автоматизація. Хоча IFTTT розширюється для роботи з багатьма датчиками та пристроями для домашньої автоматизації.

1.9.4 Amazon Echo

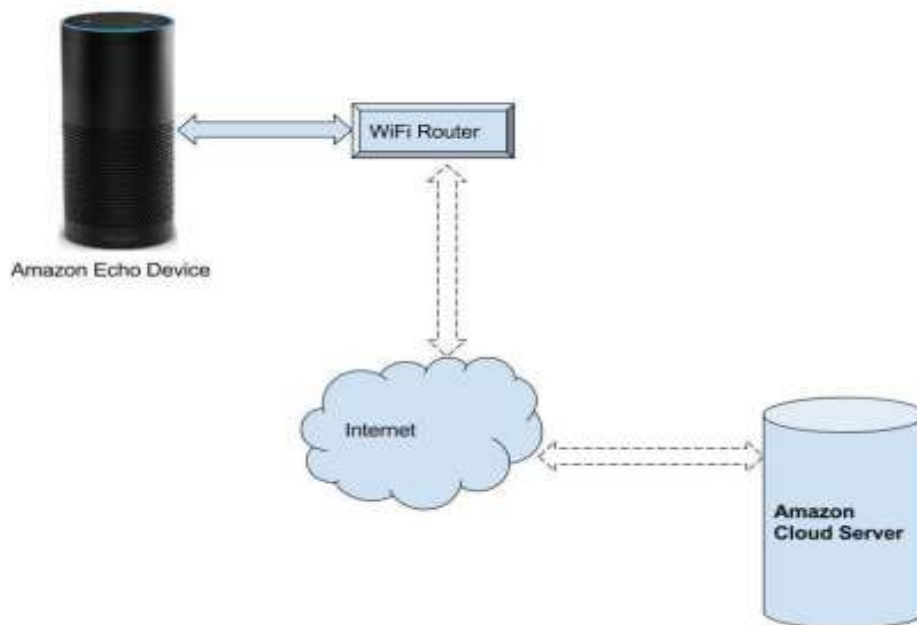


Рис 1.4. Підключення системи Amazon Echo до серверів.

Amazon – це ще один гравець на ринку з унікальним розумним домашніми пристроями, що мають назву Amazon Echo. Amazon розробляє комплексний продукт домашньої автоматизації та допомоги користувачеві для замовлення товарів з їх веб-сайту лише своїм голосом. Розробка повинна була бути достатньо розумною, щоб зрозуміти, що користувачеві потрібно, просто природним шляхом обробки голосу. У команди розробників Amazon була ідея, що обробка природної мови пристроєм могла використовуватись для більшого ніж просто замовлення. Вони придумали мінімалістичний, але інтуїтивно зрозумілий дизайн циліндричної колонки. Пристрій - це просто динамік із 7 мікрофонами, що зменшують шум. Пристрій може реагувати на голосові команди будь-якого користувача та їхнє

природне мовлення. Пристрій Amazon Echo реагує на команду пробудження. Команда - "Alexa" або "Amazon" або "Echo". Echo має різнокольорові світильники, які вказують користувачеві, що він слухає. Вони загоряються, коли пристрій прокидається і коли користувач вимовляє команди.

1.9.5 Belkin's Wemo Switch

Белкін вже мав ім'я в мережевих пристроях, пристроях живлення та кабельних аксесуарах. Белкін вийшов на ринок із перемикачем на основі Wi-Fi, перш ніж схожі рішення з'явилися в таких компаніях, як Apple, Google та Amazon. Перемикачі Wemo були запуснені в 2012 році. Тоді, перемикачі Wemo контролювалися за допомогою додатку для iPhone / iPod / iPad. У січні 2013 року Белкін оголосив про підтримку ОС Android. Вони зацікавились можливостями смартфонів, для управління домашніми пристроями. Додаток, пов'язаний з обладнанням, був тим самим, що ринок і хотів побачити. Концепція програми для смартфонів стала примхою. Белкін зміг досягти найвищих результатів у питанні "Що нового можуть робити додатки?». Перемикачі Wemo працюють з дещо іншою архітектурою в порівнянні з HomeKit від Apple, та Amazon Echo. Белкін не вважають за необхідне мати централізований пристрій керування для створення розумного домашнього середовища. Домашній комплект Apple і Amazon Echo використовують архітектуру центральних пристроїв керування для створення своєї платформи домашньої автоматизації. Перемикач Wemo використовує спеціальну IP-адресу для з'єднання з додатком для смартфонів. Кожен Wemo комутатор має вбудований Wi-Fi чіп, який може індивідуально діяти як TCP / IP-сервер. Додаток безкоштовно можна завантажити через магазини додатків. Після встановлення додаток просить користувача підключитися до власної точки доступу Wi-Fi Wemo. Потім додаток просить користувача здійснити вхід для основного ма-

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		19

ршрутизатора. Цей маршрутизатор - це маршрутизатор Wi-Fi, який підключається до вашого розумного пристрою. Тоді додаток посилає облікові дані комутатору Wemo. Перемикач Wemo використовує ці вхідні дані для входу та ідентифікує себе як пристрій бездротової мережі та отримує IP-адресу. Потім комутатор надсилає його своєму додатку, який зберігає його в пам'яті для відправки та отримання повідомлень.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		.20

Висновки до розділу 1.

У першому розділі роботи було проведено аналіз розподілу користувачів смартфонів та персональних комп'ютерів. Як ми можемо побачити зараз існує тенденція на перехід із ПК на смартфони, тому що це більш зручно. На сьогоднішній день вже більша частина інтернет трафіку проходить саме через смартфони. Саме по цій причині я дійшов висновку, що доцільно буде розробляти систему з інтерфейсом для смартфона.

Також була розглянута частина ринку інтернету речей. Разом з цим, ми дослідили та проаналізували принцип роботи та спробували зрозуміти архітектуру деяких продуктів великих компаній, таких як Google, Amazon, Belkin, Apple. Завдяки цьому ми отримали можливість зрозуміти принцип роботи систем домашньої автоматизації різних компаній. На основі цих знань ми змогли обрати рішення які більше всього нам підходять та за допомогою доповнення та повної переробки існуючих рішень зробити прототип своєї системи домашньої автоматизації.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		21

РОЗДІЛ 2.

ПРЕДСТАВЛЕННЯ СВОГО РІШЕННЯ

2.1 Вступ до домашньої автоматизації з можливістю управління голосом.

Система призначена для розуміння голосових команд користувача на природній мові разом із підтримкою сенсорного керування на Android пристрої. Програмне забезпечення системи управління можна встановити на пристрій Android як додаток. Пристрій Android повинен мати можливість підключення до Wi-Fi для з'єднання з обладнанням. Додатково також потрібна налаштована бездротова локальна мережа (WLAN), щоб система спілкувалася зі спеціально розробленим обладнанням для управління кінцевими пристроями.

У цьому розділі дипломної роботи також досліджуються конкретні технології, за допомогою яких реалізовано архітектуру розумного будинку з системою управління голосом. Тут ми обговорюємо загальний дизайн проекту та причини, що призвели до вибору конкретних технологій. Наприклад: чому замість Bluetooth або ZigBee був обраний Wi-Fi? Чому апаратне забезпечення було розроблене на платформі Arduino? Ця теза не стверджує, що в цій роботі був розроблений новий спосіб зв'язку з розумним будинком. Тут відображаються причини, чому було обрано Wi-Fi замість Bluetooth та ZigBee.

2.2 Причини використання Wi-Fi замість ZigBee та X10

Як було пояснено вище, дана система використовує Wi-Fi в якості основного засобу зв'язку. Причинами для вибору Wi-Fi є:

2.2.1 Використання

Wi-Fi - це широко розповсюджена технологія підключення різномірних типів комп'ютерних пристроїв. Це найпоширеніша технологія підключення мережі

достатнього розміру у форматі LAN, будь то настільний ПК, ноутбук, смартфон, планшет або будь-який інший комп'ютер з можливістю підключення до Wi-Fi. ZigBee і X10 - це спеціалізована мережа, для створення якої потрібні спеціальні пристрої. Наприклад: ZigBee потребує контролера ZigBee (ZC), маршрутизатора ZigBee та ZigBee. Що стосується X10, для передачі інформації йому потрібен спеціальний тип передавача-приймача лінії електропередачі в аналоговому режимі для пристроїв.

2.2.2 Вартість.

Wi-Fi - дуже дешева технологія. Якщо на комп'ютері немає бездротової карти щоб підключитися до мережі Wi-Fi, то існує велике різноманіття приладів, що можна використовувати разом з ПК. Ціни варіюються від 10 до 100 доларів. Навіть якщо немає маршрутизатора Wi-Fi, ми все одно можемо створити точку доступу за допомогою ноутбука або смартфона. На закінчення, більшість комп'ютерів (малих / великих) мають можливості підключення до Wi-Fi мережі. Але якщо комп'ютеру не вистачає обладнання для підключення до мережі Wi-Fi, надбудова апаратного забезпечення для підключення до мережі Wi-Fi дуже дешева. Навіть виділений роутер -непотрібний для створення мережі, так як сьогодні комп'ютерні пристрої досить потужні щоб створити віртуальний маршрутизатор Wi-Fi як власну захищену мережу, що працює як точка доступу.

2.2.3 Доступність.

Це одна з найважливіших причина вибору Wi-Fi над іншими типами протоколів зв'язку для домашньої автоматизації. Wi-Fi - найпоширеніша бездротова технологія, яка використовується для підключення неоднорідного комп'ютерного обладнання, що працює на неоднорідному програмному забезпеченні. Наприклад; мережа Wi-Fi може підключати ПК, Mac, машини на базі Linux, пристрої Android і це все буде працювати в одній мережі. Для доступу до мережі

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		23

комп'ютери не потребують спеціального обладнання. Технологія Wi-Fi настільки поширена, що більшість будинків обладнані нею. Таким чином, непотрібна установка спеціального обладнання для того щоб забезпечити роботу системи автоматизації розумного дому .

2.2.4 Хмарні обчислення легші за допомогою Wi-Fi.

В даний час хмарні обчислення величезні. Пристрій контролера можна використовувати для інтеграції потужних хмарних серверів для управління домашньою автоматизацією. Ось приклад – синтез мови. Додаток системи автоматизації може передавати команди, засновані на природній мові, ці команди йдуть на контроллер, який в свою чергу, використовує хмарні служби для перетворення розмовних команд у текстові, а потім додаток приймає рішення, розуміючи, які вимоги користувача.

2.2.5 Мікроконтролер Arduino.

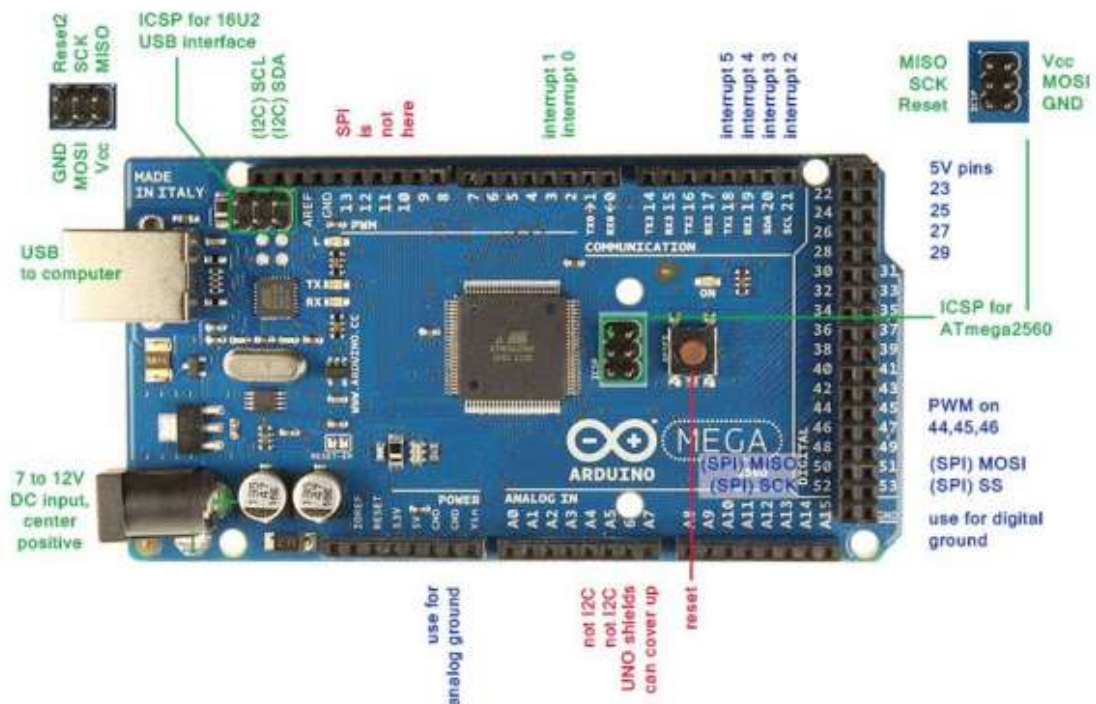


Рисунок 2.1 Arduino Mega2560

Arduino - одна з найбільш широко використовуваних платформ мікроконтролерів для прототипування. Я візьму за основу простий та надійний контролер серії Atmega від корпорації Atmel. Платформа Arduino вже має живлення, скинуте налаштування схеми, а також схему програмування та зв'язку з мікроконтролером через USB. Це робить його одним з найкращих і найпростіших для тестування та прототипування обладнання. Огляд Arduino Mega 2560 можна побачити на рисунку 2.1.

Arduino має величезну підтримку спільноти в інтернеті та має безліч документацій для кожного типу апаратного забезпечення, яке можна підключити до мікроконтролера. Arduino забезпечує простішу платформу програмування порівняно із звичайним програмуванням мікроконтролерів, де користувачі мають дуже заздалегідь заданий набір функцій та команд. Це стає набагато легше для початкових користувачів, що програмують Arduino. Крім того, плата Arduino, якій потрібно +5В для роботи, може просто житися через USB-порт комп'ютера.

Arduino поставляються в різних розмірах для гнучкої зручності використання. Варіант основної версії Arduino - це Uno з мінімальною кількістю портів та мов програмування для підтримки апаратного забезпечення, підключеного до неї. Якщо користувачеві потрібен більш потужний мікроконтролер з більшою кількістю пінів і портів, Arduino Mega може стати найкращим варіантом. Також програмний код є портативним на різних типах плат Arduino. Таким чином, код написаний для Arduino Uno, можна легко перенести на Arduino Mega без змін. Arduino дійсно дешеві, і вони легко доступні у всьому світі. Arduino Uno можна придбати за ціною менше 10 доларів. Поставляється вона в комплекті з USB-кабелем для з'єднання з комп'ютером.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		25

За допомогою програмного забезпечення Arduino IDE можна безкоштовно програмувати та завантажувати машинний код на обладнання. Програмне забезпечення Arduino просте у використанні для новачків, але досить гнучке для досвідчених користувачів. Цей додаток можна встановити на Mac, Windows та Linux. На рисунку 2.2 ми можемо побачити як виглядає додаток для розробки скетчів для Arduino на Windows.

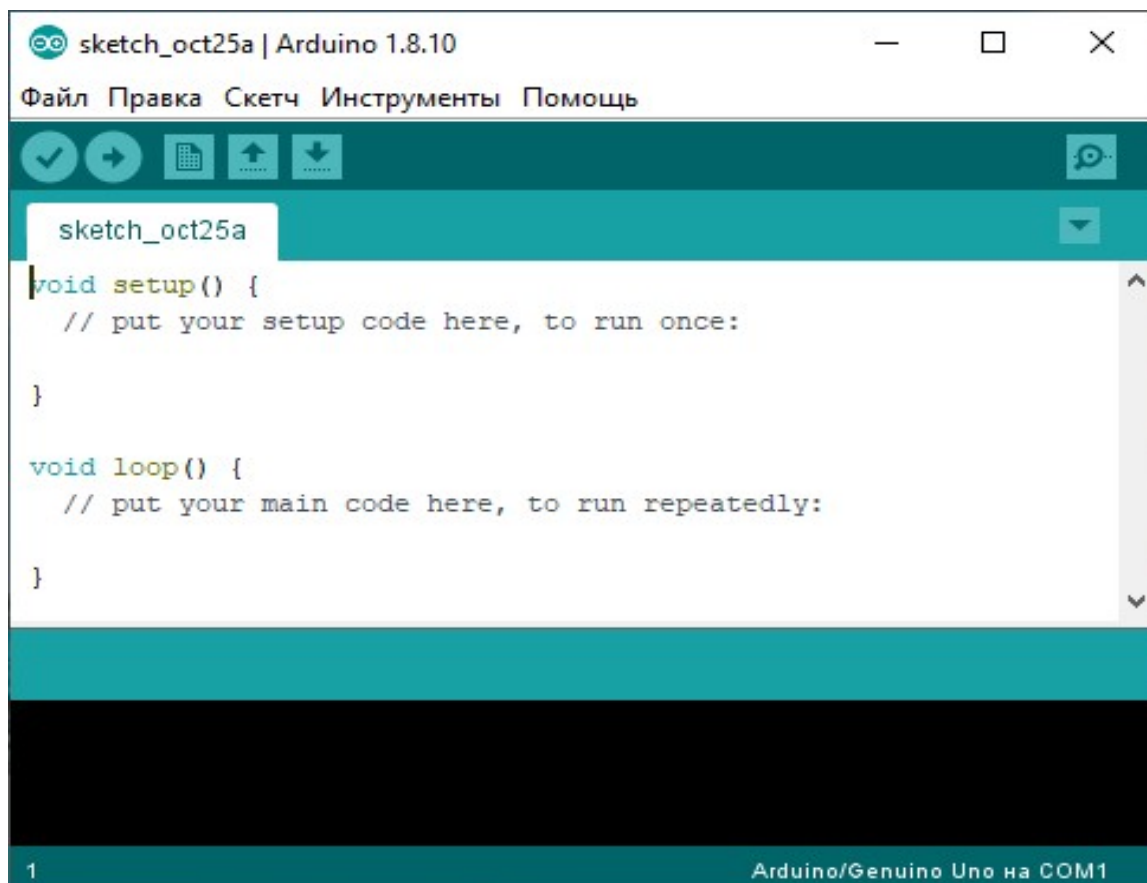


Рис 2.2. Вигляд вікна програми для розробки скетчів для Arduino.

2.3 Компоненти домашньої автоматизації.

Архітектура IVA розбита на три основні категорії, про які ми поговоримо:

1. Обладнання має можливість підключення до маршрутизатора. Воно також може вмикати / вимикати вказані пристрої, наприклад, світильники та вентилятори. Назвемо це "Блок управління".

2. GUI для користувачів та програмування програмного забезпечення, що забезпечує естетику для користувача.

3. Наш власний інтерпретатор природної мови для розуміння команд користувача, за допомогою якого відбувається перетворення з голосу в текст.

Блоків управління може бути декілька в одному будинку. Вони працюють у симбіозі та будуть контролюватися за допомогою одного додатку. Кожна панель управління матиме свою унікальну IP-адресу WLAN, утворену маршрутизатором. Ці IP-адреси використовуватимуться для надсилання та отримання повідомлень від додатку на пристрої Android. На рисунку 2.3 наведено приклад такого блоку управління.

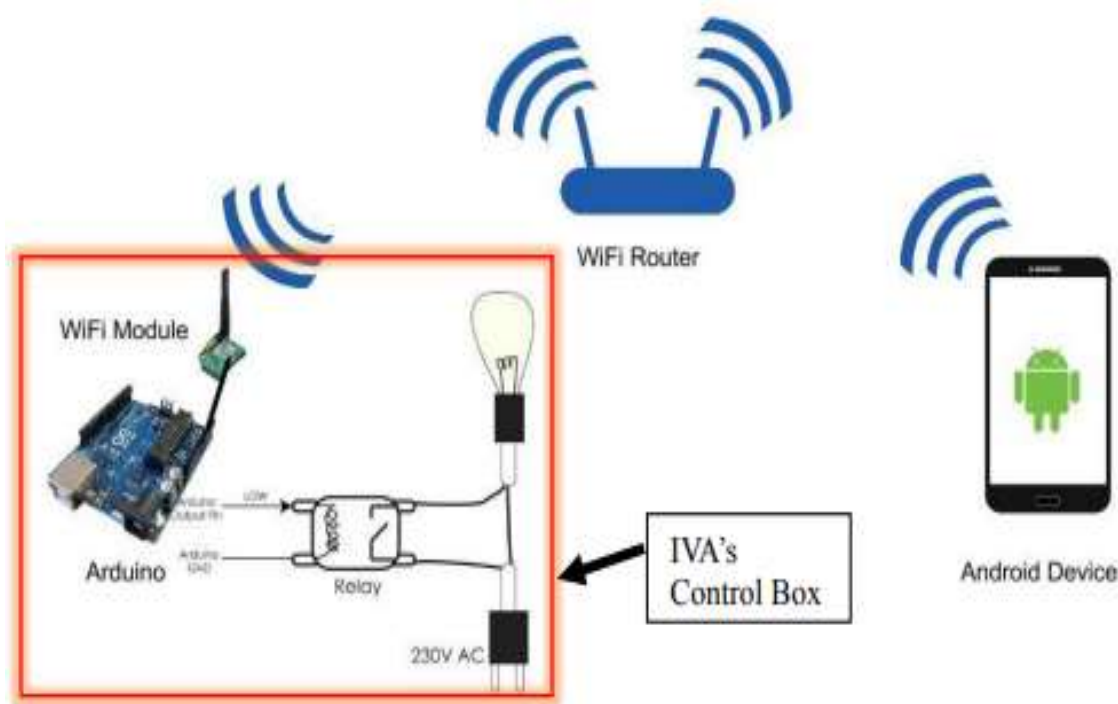


Рисунок 2.3 - Приклад блоку управління.

2.4 Блок управління IVA

IVA має мікроконтролер для інтерпретації команд та виконання завдань відповідно до команди, отриманих з мобільного додатку. Як було обговорено в раніше, використовується плата Arduino Mega 2560 як основний контролер. Ми будемо називати цю збірку пристроїв за допомогою Arduino Mega2560 як "Блок управління".

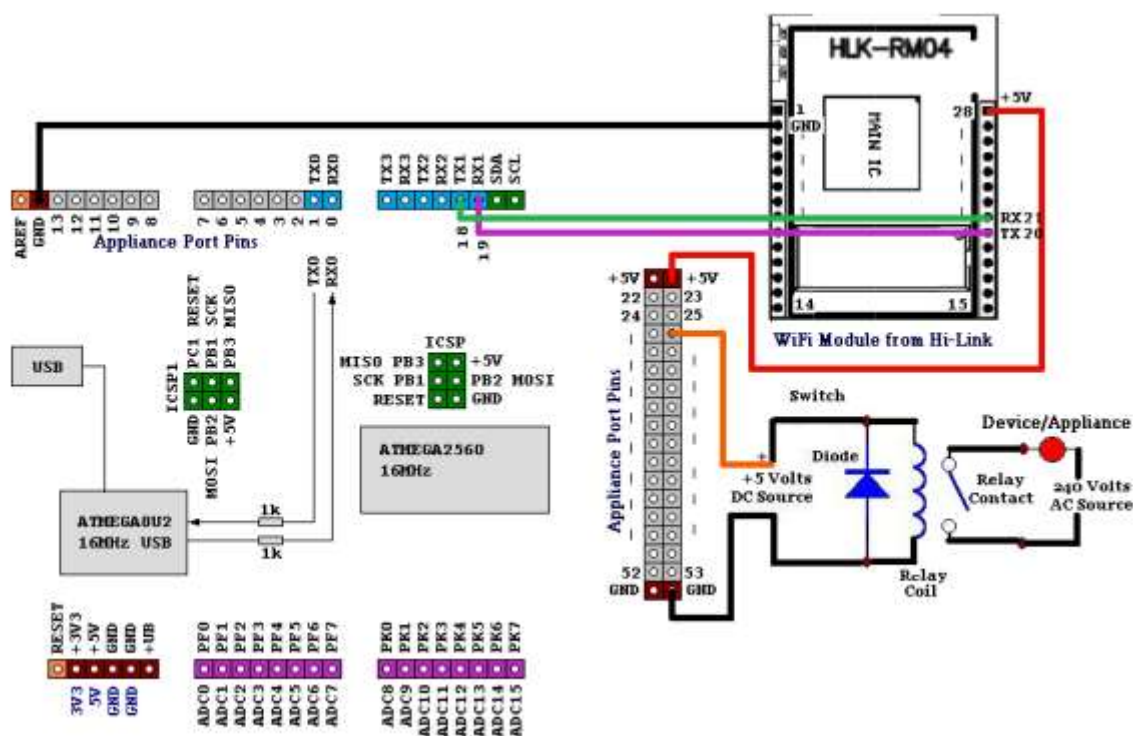


Рисунок 2.4 – Схема підключення блоку управління IVA.

2.5 Підключення пульта управління:

1. Установка електроживлення: Arduino використовує 5В як джерело живлення. Подача струму силою 1А більше ніж достатньо для того, щоб наша апаратна робота працювала. Хоча, споживання електроенергії напряду залежить від кількості пристроїв, підключених до портів.

2. Реле-перемикач, яким керуватимуть команди вводу / виводу плати. Тут, на схемі контакт 27 відповідає за керування реле вимикача.

3. Порт UART використовує контакт RX для приймання та контакт TX для передавання команд. Ми підключили Wi-Fi чіп HLK-Rm04 для надання функції бездротового зв'язку IVA.

2.5.1 Пояснення схеми.

На малюнку 2.3 зображено розгортки плати Arduino Mega 2560. На платі, контактні номери 0, 1, 14, 15, 16, 17, 18, 19 - це універсальний асинхронний приймач порту зв'язку передавача (UART). Ми будемо використовувати одну з пари 18 (TX1) та 19 (RX1) для підключення до модуля Wi-Fi.

Модуль Wi-Fi HLK-Rm04, який буде використовуватися, розроблявся компанією HI-LINK Co. Ltd. Усі інші немарковані піни, це піни для підключення 5В цифрових датчиків або інших цифрових пристроїв. Піни тут використовуються для управління кінцевими приладами. Таким чином, піни вводу/виводу будуть підключені до перемикачів контролерів кінцевих пристроїв за допомогою реле або регуляторів напруги.

Для управління включенням/вимкненням пристроїв буде використовуватися релейний вимикач. Релейний перемикач зазвичай використовується для управління високовольтними пристроями, такими як світильники, вентилятори тощо. Контролер вводу / виводу в основному має вихідну напругу 5В.

2.5.2 Схема живлення.

На рисунку 2.3 розетка живлення позначена міткою. Підключення постійного струму 5 В буде жити плату Arduino. Потім до Arduino я підключу Wi-Fi чіп HLK-RM04, за допомогою VCC шпильки GND. Будь-який з трьох + 5 В контактів можна підключити до піну 28 чіпа Rm04. Аналогічно, пін заземлення (GND) з плати Arduino повинен бути підключений до модуля Wi-Fi контактом

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		29

номер 2. Це забезпечить живлення плати Arduino, яка, в свою чергу, буде жити чіп Wi-Fi.

2.6 Підключення та налаштування Wi-Fi.

Піни 20 і 21 модуля HLK-Rm04 Wi-Fi підключені до пінів Arduino Mega 18 і 19 для передачі (TX) і прийому (RX) відповідно. Пін TX модуля Wi-Fi з'єднаний з RX-піном Arduino, так що лінія передачі даних підключена до лінії прийому даних протилежного контуру і навпаки. Модуль Wi-Fi повинен бути налаштований таким чином, щоб він міг з'єднуватися з уже наявними бездротовими мережами. Кожному блоку управління потрібен один модуль Wi-Fi для ідентифікації його в мережі. Більше того, це збільшує доступ ІВА до іншого обладнання в інших приміщеннях.

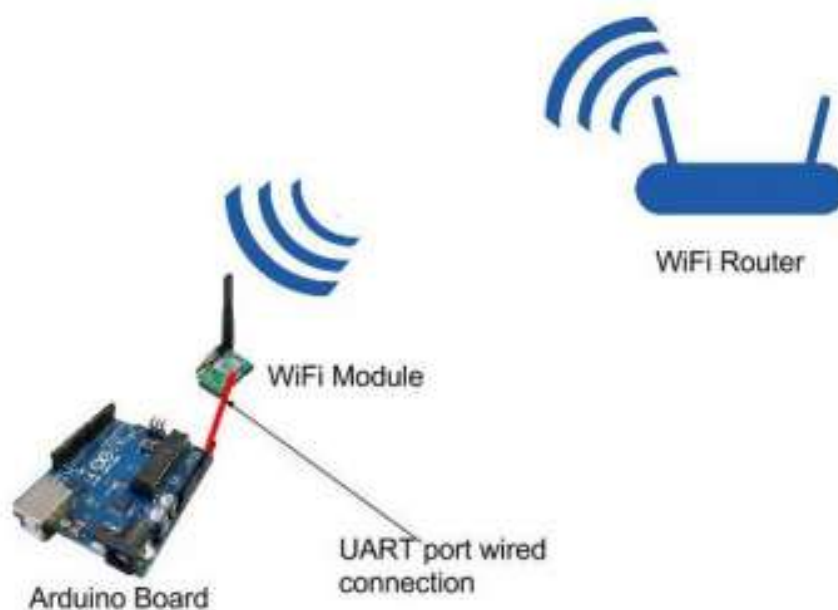


Рис. 2.5. Підключення Arduino до мережі WiFi.

Wi-Fi модуль блоку управління повинен бути налаштований як TCP / IP-сервер у мережі. Таким чином кожен модуль на базі HLK-Rm04 матиме уніка-

льну IP-адресу. Тоді смартфон може підключитися до сервера, який виступає клієнтом. IP-адреси сервера встановлюються в додатку IVA заздалегідь через екран налаштувань.

Плата Arduino підключена до мікросхеми Wi-Fi через UART порт за допомогою проводів. Чіп HLK-Rm04 HiLink має багато режимів роботи. Налаштування відбувається за допомогою маршрутизатора. Він може діяти як клієнт після з'єднання з маршрутизатором, він може діяти як точка доступу, і стати самим міні-роутером. Але режим, який ми будемо використовувати - це налаштування контроллера як сервер із виділеною IP-адресою по WLAN мережі. Мікросхема HLK-Rm04 має заздалегідь заданий стек для того, щоб підключитися як TCP / IP сервер та спілкуватися по мережі через простий текст ASCII. IVA використовує самостійну установку зв'язку TCP / IP для з'єднання з контрольним входом від програми.

2.8 Фізичні з'єднання

Підключаємо модуль Wi-Fi HLK-Rm04 до плати Arduino. Після цього підключаємо роз'єм 5В постійного струму живлення на платі Arduino Mega. Після того, як всі з'єднання на місці, увімкнено плату Arduino, яка працює з модулем Wi-Fi.

Тепер нам спочатку потрібно встановити модуль Wi-Fi як TCP / IP-сервер та його унікальну IP-адресу. HLK-Rm04 повинен бути підключений до комп'ютера за допомогою кабелю Ethernet. Провідне з'єднання буде відображатиметься на ПК. Потім із браузера перейдіть за посиланням: <http://192.168.16.254/ser2net.asp>. Це сторінка, що показує активні серверів за замовчуванням для налаштування модуля Wi-Fi. Користувачеві буде запропоновано ввести облікові дані. За замовчуванням, ім'я користувача - "admin", а пароль

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		31

- "admin". Переходимо на екран, подібний до рисунку 2.5 та заповнюємо всі потрібні нам поля як показано нижче.

2.9 Налаштування параметрів.

The screenshot shows the 'HLK-RM04 Serial2Net Settings' page. On the left is a navigation menu with options like 'Serial2Net Settings', 'Advance Settings', 'Serial2Net UART 2 Settings', and 'Administration'. The main area contains the following settings:

- NetMode:** A dropdown menu set to 'WIFI(CLIENT)-SERIAL'.
- SSID:** An input field with a 'Scan' button next to it.
- Encrypt Type:** A dropdown menu set to 'WPA2 AES'.
- Password:** An input field.
- IP Type:** A dropdown menu set to 'DHCP'.
- Serial Configure:** A table with two columns: 'Current' and 'Updated'. The 'Current' value is '250000,8,n,1' and the 'Updated' value is '115200,8,n,1'.
- Serial Framing Lenth:** A table with two columns: 'Current' and 'Updated'. The 'Current' value is '2' and the 'Updated' value is '64'.
- Serial Framing Timeout:** A table with two columns: 'Current' and 'Updated'. The 'Current' value is '0 milliseconds' and the 'Updated' value is '10 milliseconds (< 256, 0 for no timeout)'.
- Network Mode:** A dropdown menu set to 'server'.
- Remote Server Domain/IP:** A table with two columns: 'Current' and 'Updated'. The 'Current' value is '192.168.11.245' and the 'Updated' value is '192.168.11.245'.
- Local/Remote Port Number:** A table with two columns: 'Current' and 'Updated'. The 'Current' value is '8080' and the 'Updated' value is '8080'.
- Network Protocol:** A dropdown menu set to 'TCP'.
- Network Timeout:** A table with two columns: 'Current' and 'Updated'. The 'Current' value is '0 seconds' and the 'Updated' value is '0 seconds (< 256, 0 for no timeout)'.

At the bottom right of the settings area are 'Apply' and 'Cancel' buttons.

Рисунок 2.6 – Налаштування Wi-Fi модуля.

- **NetMode:**
Виберіть NetMode як "WIFI (SERVER) -SERIAL". Це налаштує Wi-Fi модуль як сервер, до якого можна отримати доступ, використовуючи власну IP-адресу.
- **SSID:**
SSID - назва мережі Wi-Fi, яка буде мережею маршрутизації. WLAN - це назва домашньої мережі, до якої підключено смартфон та ПК.

- **Тип шифрування:**

Тип шифрування залежить від маршрутизатора або комутатора. Щоб дізнатись тип шифрування у вашій мережі, комп'ютер спочатку повинен бути підключений до WLAN.

- **Пароль для модуля Wi-Fi:**

Пароль для мережі.

- **Тип IP:**

Вибрано тип STATIC IP. Це показує, що йому потрібна конкретна IP-адреса, як і розумним клієнтським пристроям (телефонам) потрібні точні IP-адреси, що не змінюються, для спілкування з обладнанням.

- **Конфігурація послідовного порту:**

Серійний порт з'єднує плату Arduino з мікросхемою модуля RM04. Для налаштування послідовного порту, і Arduino, і чіп Wi-Fi повинні мати однакові значення параметрів. Тут використовуються конфігурації:

o Serial configure = 9600,8,n,1

o Serial Framing Length = 64

o Serial Framing Timeout = 10

- **Режим мережі:**

Мережа вибрана як Сервер. Це режим TCP / IP. Тут показано, що Wi-Fi чіп буде виконувати роль TCP / IP-сервера при підключенні до мережі.

- **Порт:**

Номер порту за замовчуванням зберігається незмінним. Це 8080.

Тепер, коли Wi-Fi модуль HLK-Rm04 підключений до мережі, додаток IVA на пристрої Android може надсилати команди у відповідне поле керування. Додаток Android буде діяти як клієнт і відкриває потік з'єднання з сервером. Модуль Wi-Fi, який ми зараз налаштовуємо як сервер у мережі прийме з'єднання, а потім

отримає команду з додатку. Потім команда інтерпретується, тобто розбивається на різні поля щоб зрозуміти, що модуль повинен робити.

2.10 Отримання команди з програми.

Блок управління отримує команду у вигляді тексту на своєму порту UART. Це послідовний порт, який підключає модуль Wi-Fi HLK-Rm04 до плати контролера Arduino. З'єднання портів можна побачити на схематичному рисунку 2.3. Програма на контролері Arduino постійно проводить опитування даних про порт UART. Таким чином, якщо дані є, вони перевіряють отриману команду. Для активації порт UART в Arduino використовується наступний код: `Serial1.begin(9600);`

Код опитування показаний нижче:

```
while (!Serial1) { ;  
// wait for serial port to connect. Needed for native USB  
}
```

2.11 Структура команд.

Команда складається з одинадцяти символів. Більшість символів мають числовий характер. Давайте розберемо команди на складові частини.

Спочатку команда інкапсульована символами старту та зупинки. Будь-яка команда повинна починатися з символу зірочки (*) і закінчуватися символом решітки (#). Усі символи, крім символів початку (*) та стопу (#) мають числовий характер.

«*» Супроводжується чотирма цифровими символами числа унікальної домашньої ідентифікації. Ці чотиризначні символи унікальні для дому. Це ро-

битися для того щоб лише авторизований пристрій Android між керувати обладнанням. Наприклад, '* 1234XXXXX #'. Тут домашній ідентифікаційний номер - 1234. Усі символи "X" - це наступна частина команди.

Шоста і сьома цифри ідентифікують номер кімнати. Таким чином, IVA може підтримувати до 99 унікальних кімнат для дому. Наприклад, команда до ідентифікаційного номеру кімнати буде: '* 123401XXX #'. Тут, кімната адресується як 01.

Наступні два числа - це ідентифікаційний номер пристрою. Ці двозначні числа інформують яким пристроєм буде керувати мікроконтроллер.

Наприклад, команда буде мати вигляд '* 12340102X #'. Команда показує, що унікальний домашній номер - 1234, номер кімнати - 01, а пристрій, який має контролюватися - 02.

Другий останній символ, тобто символ перед символом зупинки (#) – це інформація про пристрій управління. Тут мікроконтролер запрограмований з операторами корпусу комутатора для виконання дій над пристроєм. Наприклад, якщо код пристрою для включення дорівнює 1, то вся команда з додатку для Android буде '* 123401021 #'. Це означає, що будинок номер 1234, кімната 01, пристрій 02 потрібно включити.

2.12 Контроль фізичних пристроїв.

У плати Arduino багато пінів вводу / виводу. Ці піни використовуються для передачі напруги 5В по лінії до підключених пристроїв. Arduino - дуже сильна плата контролера. Вона має попередньо побудований механізм живлення. Можна підключати живлення низької напруги прямо через пін. Це може забезпечити струм 40мА на пін вводу / виводу при його витягуванні. Таким чином, немає потреби в постачанні окремого джерела живлення для таких компонентів, як світ-

лодіоди, звукозаписуючі сигнали тощо. I/O пін, Arduino повинен бути запрограмований з номером контакту як вихідний порт. Наприклад, якщо пін номер 5 потрібно підтягнути, необхідно записати рядок у програмі - щось на зразок pinMode (5, OUTPUT). Тут контактний номер 5 буде виступати як вихідний штифт і буде внутрішньо підтягнуто. Так, наприклад, коли блок управління отримує команду для увімкнення світла, мікроконтролер повинен зробити контактний номер 27 високим. Це тому, що реле-перемикач підключається до мікроконтролера через контактний номер 27. Ми можемо бачити це на схематичному рисунку 2.3. Рядок у коді, щоб подати напругу на 27 пін в Arduino буде виглядати так - " digitalWrite (27, HIGH)".

2.13 Фрагмент коду методу "налаштування" для панелі керування

```
1. void setup()
2. {
3.   int switch=27, index=0;
4.   Serial1.begin(9600);
5.   Serial2.begin(9600);
6.   pinMode(ledPin, OUTPUT); // sets the digital pin as output for switching
44
// Pins 22, 24, 26 and 28 act as the controlling speed (regulating) the AC power voltage
//
regulation
7.   pinMode(22, OUTPUT);
9.   pinMode(24, OUTPUT);
10.  pinMode(26, OUTPUT);
11.  pinMode(28, OUTPUT);
```

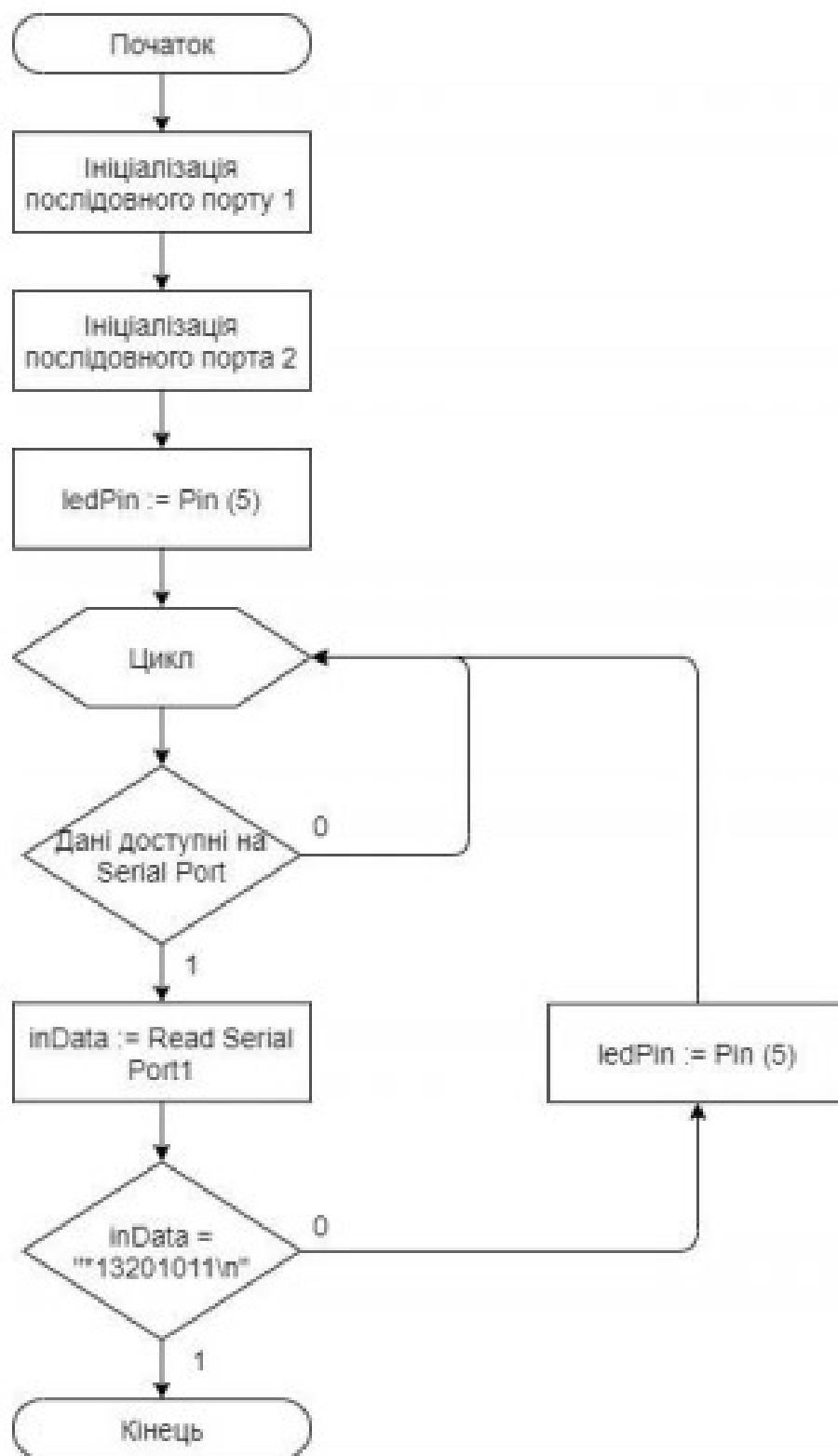


Рис. 2.7. Блок-схема послідовного циклу.

Метод налаштування є обов'язковим методом у програмі Arduino. Програма Arduino не буде працювати, якщо метод установки не запрограмований. Метод налаштування запускається вперше коли мікроконтролер завантажується або перезавантажується. Це робить його дуже простим для нових програмістів, щоб визначити все, що потрібно ініціалізувати. Наприклад, штифти, що використовуються, їх напрямки, кількість портів та їх швидкість передачі даних, ініціалізація різних змінних. У вищеприведеному способі рядки 4 та 5 мають налаштувати послідовні порти на Arduino Mega. Послідовний порт 1 підключений до чіпа Wi-Fi HLK-Rm04. Пін, який буде виступати в якості сигнального до реле буде налаштовано як вихідний контакт. Це означає, що контактний сигнал буде відправляти 5V сигнал / команду на реле вимикача. Тут контактний номер 27 встановлений як вихідний пін, який потім сигналізує перемикачу реле або про включення, або відключення. Якщо пін має + 5 В, це сигналізує реле підключено. Якщо штифт 0В, пристрій вимикається. Більше того, як пояснено в коментарях до програми, пінами також є цифри 22, 24, 26 і 28 що встановлюються як вихідні піни для управління регулятором напруги живлення.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		38

Висновки до розділу 2.

У другому розділі були вивчені та проаналізовані різні типи бездротового підключення для домашньої автоматизації. В даній роботі ми будемо використовувати саме WIFI як основний тип підключення. Завдяки цьому ми зможемо отримати відносну простоту налаштування системи, невелику ціну та швидкість підключення, що повністю задовольняє наші потреби.

Разом з цим було виконане підключення модуля WIFI та його подальше налаштування для потреб системи. Ми використовуємо TCP/IP сервер для нашої системи.

Також було проаналізовано платформи на яких можна побудувати систему домашньої автоматизації з голосовим управлінням. Зупинилися ми на Arduino Mega. Це дозволяє в повній мірі виконати всі завдання, що перед нами стоять.

Також ми проаналізували як виконується підключення пристроїв до Arduino та які піни використовуються для посилення та отримання сигналу на пристрій та для живлення.

Після цього було створено систему для подання команд на Arduino та на пристрій котрим ми керуємо. В закодованій команді "* 123401021 #" можна зрозуміти що в будинку №1234 в кімнаті 1 потрібно увімкнути пристрій 2. Таким чином число кумнат що підтримуються може бути 99 та пристроїв у кожній кімнаті може бути також 99.

Після цього було розібрано як підключити та налаштувати Arduino з програмного боку. Як можна побачити з цим не виникне проблем навіть у програміста початківця, тому, що для програмування Arduino використовується мова програмування «C».

Розділ 3.

Програмне забезпечення та інтеграція IVA

3.1 Інтерфейс користувача.

IVA має дуже загальний інтерфейс користувача, розроблений у XML. Для переключення є дві кнопки увімкнути / вимкнути пристрої, в основному лампу або кавову машину. Є ще одна кнопка, яка активізує можливість голосового прослуховування IVA через мікрофон пристрою Android. Ця кнопка розміщується внизу екрана. Є повзунок управління або смуга треку управління в C#. Android називає це "смугою пошуку". Панель пошуку Android керує напруженою регулювання регулятора, підключеного до блоку управління. Всі графічні компоненти та інтерфейси користувача Android розроблені в XML. Реалізація компонентів інтерфейсу та завдань, які вони виконують, обробляються кодом Java. Android не використовує традиційний код Java Swings для управління елементами управління. Android надає власні механізми, методи, слухачі та події для взаємодії з графічними компонентами. Екран в Android називається "активністю". Додаток повинен мати графічний інтерфейс та зв'язаний з ним XML-файл для відображення екрана користувачеві. Екран активності повинен мати базовий контейнер, який називається "макет". Макет в Android може містити багато компонентів і може влаштувати його відповідно до потреб користувача. Макет також може містити інші дочірні макети що мають назву компоненти.

3.2 XML код.

XML code

activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/RelativeLayout1"
```

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		40

```

android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical">
<ToggleButton
android:id="@+id/toggleButton1"
android:layout_width="102dp"
android:layout_height="110dp"
android:background="@drawable/light_off"
android:text="LIGHTS(OFF)"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:textColor="#ff232323"
android:longClickable="false"
android:layout_alignParentTop="true" />
<ToggleButton
android:id="@+id/toggleButton2"
android:layout_width="140dp"
android:layout_height="110dp"
android:background="@drawable/light_off"
android:text="COFFEE"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:textColor="#ff232323"
android:longClickable="false"
android:layout_alignParentTop="true" />

<TextView

```

```

android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:text="@string/voice_box"
android:textColor="#ffff2c25" />
<Button
android:id="@+id/button1"
android:layout_width="48dp"
android:layout_height="48dp"
android:background="@drawable/mic"
android:layout_above="@+id/textView1"
android:layout_centerHorizontal="true" />
<SeekBar
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/regulator1"
android:layout_marginTop="36dp"
android:layout_below="@+id/toggleButton1"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:max="5"
android:layout_toStartOf="@+id/textView"
style="@style/AppBaseTheme" />
</RelativeLayout

```

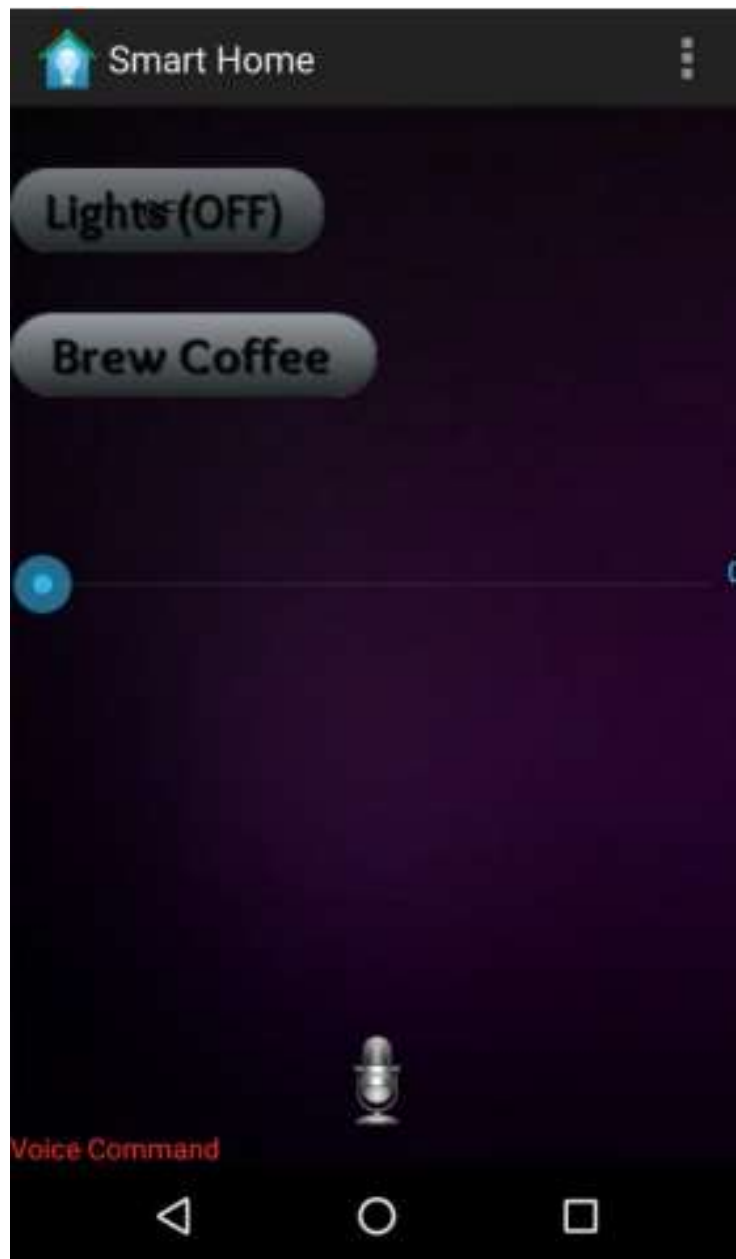


Рис 3.1 Скріншот інтерфейсу користувача IVA.

3.3 Пояснення.

Як показано в коді вище, є дві кнопки. Одна кнопка призначена для переключення пристрою. Інша кнопка використовується для запуску мови в текст Методи розпізнавання намірів. Коли натискається кнопка мовлення, вона приймає

					ІАЛЦ.467100.003 ПЗ	Арк
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

голос на вхід і виводить текст у текстовому форматі. Існує також "панель пошуку", яку можна змінити, щоб надсилати команди для управління напругою апаратною стороною.

3.4 Кнопка перемикання.

Кнопка перемикання в Android має два стани. Перший, що натиснутий - "справжній" стан. Наступне натискання – це "помилковий" стан. Подія натискання запускається при натисканні кнопки. Потім код перевіряє поточний стан кнопки перемикання та надсилає певну строкову команду в мережі.

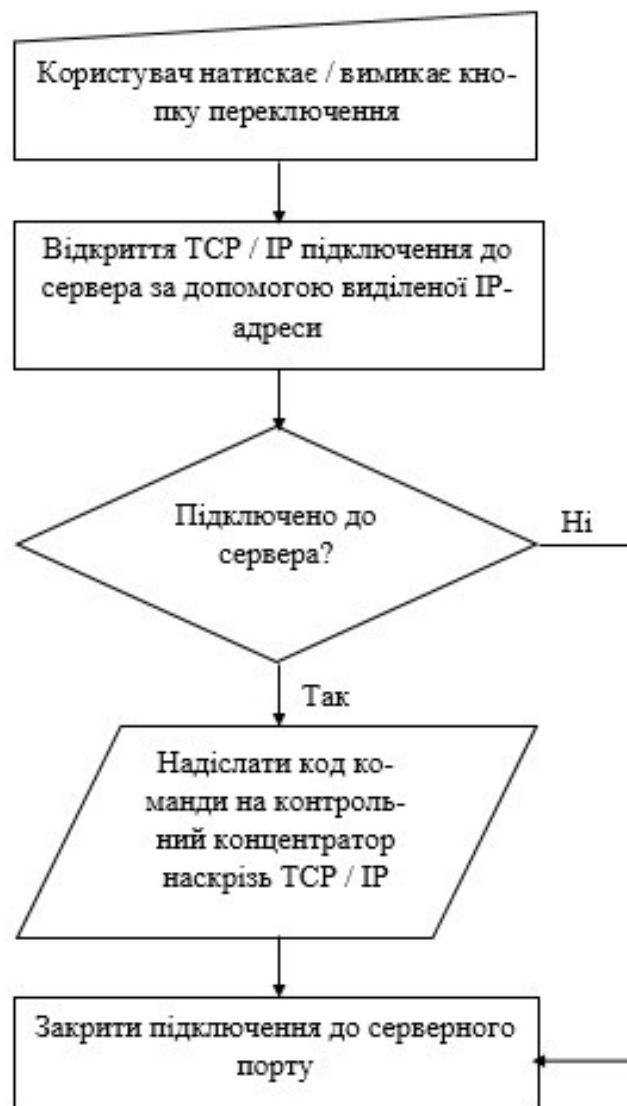


Рис. 3.2. Блок-схема для прослуховувача подій кнопки.

3.5 Активація мікрофона для голосових команд.

У нижній частині екрана є кнопка із зображенням мікрофона. Коли натискається кнопка, IVA негайно переходить у режим прослуховування. Користувачу буде запропоноване діалогове вікно команд Google Voice.

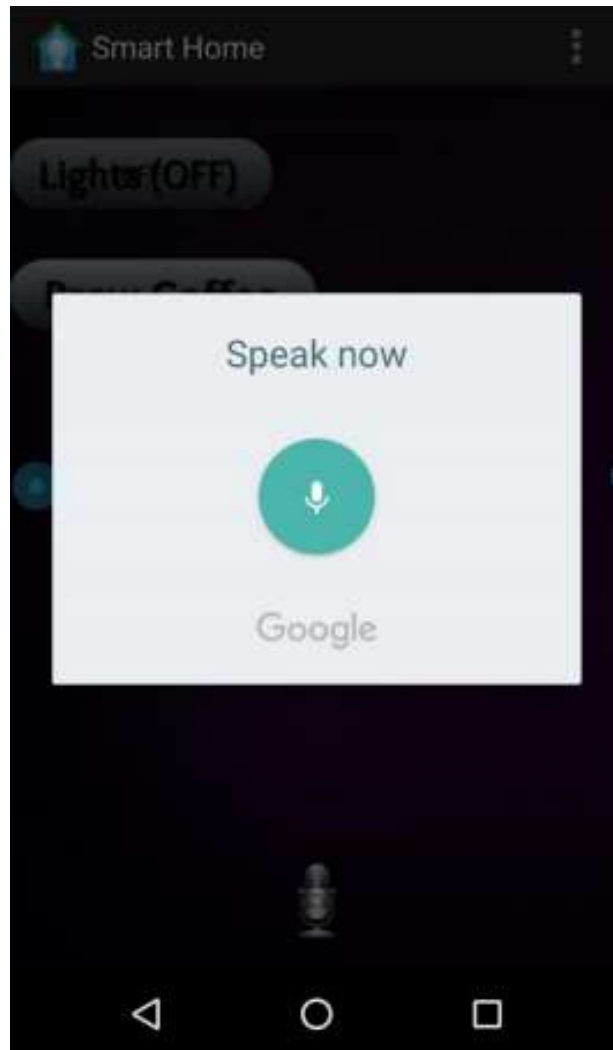


Рис. 3.3. Скріншот діалогового вікна для перевodu голосу в текст.

Після відображення вищезазначеного екрана користувач може почати говорити звичайною мовою. Потім система перетворює голос у текст за допомогою потужних хмарних серверів призначених для перетворення голосу в текст. Ми побачимо деталі перетворення мови в текст далі в. Як тільки програма отримує відповідь у текстовому форматі від серверів для обробки голосу Google , текст

відображається в текстовому режимі на екрані Android. Приклад цього можна побачити на рисунку 3.4.

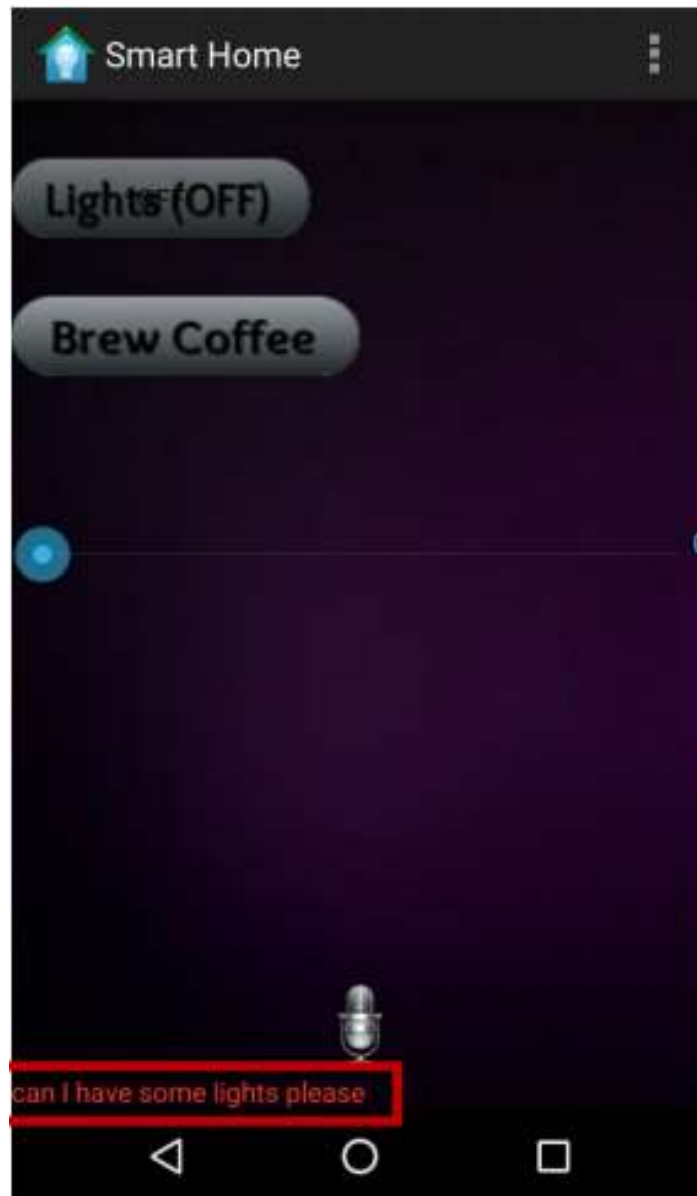


Рис. 3.4. Скріншот перетворення голосу в текст.

Як тільки IVA має текстовий формат, він намагається знайти ключові слова. Ключові слова є актуальними словами, які вже зберігаються в його пам'яті як база даних. Таким чином, база даних ключових слів може бути завантажена або додана, щоб IVA зрозумів, що робити. Як тільки IVA відфільтрував ключове

слово, він користується дуже базовим підходом до розгляду заперечення, схвалення, як так/ні тощо. Потім IVA генерує відповідну команду відповідь, яку надіслає на апаратну сторону через з'єднання Wi-Fi TCP/IP.

Таблиця 3.1. Ключові слова.

Слова	Переведено в команду	Номер кімнати	Номер пристрою	Дія
Ввімкнути світло, ввімкнути лампу, занадто темно	*123401011#	01	01	Увімкнути лампу\світло
Вимкнути лампу, вимкнути світло, занадто світло.	*123401010#	01	01	Вимкнути лампу\світло
Змінити, регулювати, зробити менше, зробити більше, швидкість вентилятора, інтенсивність, один, два, три, чотири, п'ять	*12340102(X)# Де X може бути 1,2,3,4,5	01	01	Збільшити\зменшити швидкість(напругу) регулятора пристрою

У таблиці 3.1 наведено перелік ключових слів, які IVA використовує для фільтрації та розуміння вимог користувача. Слова не є статичними для програми, їх можна додати в базу даних. IVA може прочитати їх та створити фільтр відпо-

відно до програми. Це дає особисті переваги та свободу для використання лексики користувачем за своїм смаком. Будь-яку кількість ключових слів можна додати для збільшення ефективності фільтру програми.

3.6 Конвертація мови в текст.

Найважливіший фрагмент програмного забезпечення IVA - це інтерпретація природної мови від користувача. Для перетворення розмовних слів у текст використовується API мови в текстових API, наданих Google. Додаток використовує мікрофони на пристрої Android для запису мови в аудіофайл. Потім файл надсилається на сервери для обробки мови Google. Сервери обробляють аудіо файл та генерують текстову відповідь на пристрої Android через Інтернет. Таким чином, IVA потребує Інтернету, щоб підключитися до серверів Google для надсилання аудіофайлу, записаного ним через мікрофон.

3.6.1 Обробка мовлення.

Процес перетворення мови в текст трохи складніший, ніж просто порівняння вхідного голосу із збереженими шаблонами. Як тільки аудіо буде надіслано на спеціальні хмарні сервери Google, він перетворює його з аналогового в цифровий формат. Для цього сервер бере невеликі зразки, або оцифровує звук шляхом точних вимірювань хвилі з частими інтервалами. Потім система фільтрує цей відібраний звук, щоб видалити небажаний шум, а іноді розділити його на різні смуги частоти (частота - це довжина хвилі звукової хвилі, почута людиною як різниця в тоні). Він також використовується для регулювання змін гучності, оскільки люди не завжди говорять з однаковою швидкістю та гучністю. Таким чином, звук повинен бути відрегульованим відповідно до швидкості зразків звукових шаблонів, які вже зберігаються в пам'яті системи.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		48

Далі сигнал ділиться на невеликі сегменти, короткі на кілька сотих секунди, або навіть тисячні частки у випадку приголосних звуків - зупинки приголосних, що виробляються перешкоджають потоку повітря в голосовому тракті - як "п" або "т". Потім програма відповідає цим сегментом на відомі фонем відповідною мовою. Фонема – найменший елемент мови - подання звуків, які ми видаємо і складаємо разом змістовні вирази. За даними експертів, в англійській мові існує приблизно 40 фонем мови, хоча різні лінгвісти мають різну думку щодо точної кількості. Наступним кроком є фактичні поточні дослідження, щоб зробити суперпотужні комп'ютери для обробки та аналізу вимовлених слів до точного тексту. Існують різні способи зробити це. Для прогнозування слів за допомогою пошуку використовуються різні моделі статистичного аналізу відповідних шаблонів в поданих даних. Здатність знаходити візерунок навіть для різних акцентів і діалектів здійснюються за допомогою алгоритмів машинного навчання, таких як регресії, аналіз даних та багато інших. Більше інформації можна знайти в багатьох наукових роботах з даної теми. Потім сервер знаходить відповідні звукові конструкції слова з фонетикою в базі даних та надсилає у відповідь ці слова у текстовому форматі.

3.7 Код підключення та комунікації сокета.

```
void connectAndSend(String cmd){
try {
    InetAddress serverAddr = InetAddress.getByName(SERVER_IP);
    socket = new Socket(serverAddr, SERVERPORT); // EstablishingConnection
    String str = HomeId+cmd; // Command example (01011)
    // append command to home ID Room(01) – Device (01) Action(1)
```

```

PrintWriter out = new PrintWriter(new BufferedWriter(
new OutputStreamWriter(socket.getOutputStream())),
true);

out.println(str); // Sending actual data

socket.close(); // Close open connection

} catch (UnknownHostException e1) {

e1.printStackTrace();

} catch (IOException e1) {

e1.printStackTrace();

}

} // End method connectAndSend()

```

Цей код показує з'єднання з сервером. SERVER_IP - це рядок. IP-адреса - це конкретна мережа, адреса WLAN мережі управління, вже запрограмована в додатку. Раз з'єднання встановлено, канал передачі потоку використовується для передачі інформації в мережі.

3.8 Код фільтрації ключових слів.

```

void filterHotWords(String line) {

//Filter using if() statement

if (line.toLowerCase().contains("lamp") || line.toLowerCase().contains("light") ||

line.toLowerCase().contains("bulb") || line.toLowerCase().contains("lights") ||

(line.toLowerCase().contains("dark"))) { // AND (&&)

```

```

if (line.toLowerCase().contains("so") || line.toLowerCase().contains("too") ||
line.toLowerCase().contains("on") && !line.toLowerCase().contains("not")) {
{
connectAndSend ("01011"); //Send command to witch on Lights
}
} //End method

```

У цьому коді невеликий приклад того, як IVA фільтрує ключові слова з користувальницької розмовної команди. Цей фільтр, метод *HotWords*, виконується лише після того, як відповідає сервер Google та надсилає голосові команди що перетворені у текстовий формат. Потім цей текст передається для фільтрування важливого/ключового слова, щоб зрозуміти, чого хоче користувач. Якщо конкретні слова знайдені в потрібному порядку, викликається метод *connectAndSend*.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		51

Висновки до розділу 3.

В третьому розділі наша увага була зосереджена на інтерфейсі користувача та на роботі системи для перетворення мови в текст. Додаток для Android був написаний з використанням мов XML та Java. В цьому додатку можна побачити кнопки для увімкнення та вимкнення пристроїв та кнопку для диктування тексту голосом, для подальшого його перетворення в текстовий формат на серверах Google.

Після того, як ми отримуємо команду в текстовому форматі, ми виділяємо ключові слова за допомогою яких блок управління розуміє яку команду для виконання дати на пристрій. Всі ключові слова знаходяться в базу даних, яку користувач може доповнювати та змінювати за власним бажанням. В подальшому буде можливість зробити графічний інтерфейс для цієї бази даних, щоб додавання чи зміна ключових слів стали ще легшими для кінцевого користувача.

Також був написаний метод для фільтрації та вибору ключових слів із тексту, що ми отримуємо з серверів Google після його перетворення в текстовий формат. Все це відбувається на програмування Java.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		52

Розділ 4

Висновки та розвиток системи в майбутньому.

4.1 Висновки.

Метою даної дипломної роботи було вивчення, припущення, оцінка та реверсування існуючих систем з метою розробки та реалізації власної системи домашньої автоматизації. Ця мета була успішно виконана оскільки на дизайн IVA сильно впливали вивчені архітектури домашньої автоматизації. Мережева архітектура нагадує близьку до перемикача Wemo Belkin, тоді як на голосові команди та природну обробку мови впливає дизайн Amazon Echo. Ідея пристрою в кімнаті була включена в архітектуру IVA, за допомогою розуміння, як це зроблено в термостаті Nest. Таким чином, робочий прототип архітектури домашньої автоматизації Intelligent Voice Activated (IVA) був розроблений та протестований на основі комбінації кількох систем домашньої автоматизації, спроектованих та розроблених в найкращих інноваційних програмних та апаратних компаній, таких як Google, Amazon, Apple та Belkin.

4.2 Майбутній розвиток системи.

Майбутня сфера застосування IVA може бути величезною. Є кілька ідей, які зроблять IVA більш потужним, розумним, масштабованим, а також система стане кращою для домашньої автоматизації.

4.2.1 Поліпшення захоплення голосу.

Мікрофон пристрою Android можна замінити на мікрофон, що більш чутливий для скасування високого шуму, розташований у кожній кімнаті будинку. Тоді, мікрофони можуть бути підключені до пристрою Android через радіочас-

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		53

тоти Bluetooth або з'єднання Wi-Fi. Це може усунути необхідність користува-
чеві мати пристрій Android поруч із ним, для того щоб давати голосові ко-
манди.

4.2.2 Масштабованість та додавання нових пристроїв.

Оскільки платформа Arduino відрізняється високою гнучкістю і може під-
ключатися до більшості обладнання, яке сумісне з мікроконтролером, приєд-
нання або розробка більшої кількості пристроїв, з якими працює система IVA, не
створює жодних проблем. Тип та кількість пристроїв, які можуть бути сумісна з
IVA величезна. Наприклад, біометричну здатність можна надати шляхом приєд-
нання датчик відбитків пальців до блоку управління. Іншим прикладом може
бути додавання моторизованих пристроїв таких як контролер жалюзі для вікон,
контролер дверей гаража та багато інших пристроїв. Пристрої можуть бути роз-
роблені з загальною архітектурою Arduino для того, щоб вона працювала з IVA.
Оскільки весь контроль пристроїв низького рівня здійснюється апаратом Control
Box, додати пристрій підключення та команду для додатка IVA - це зовсім не
проблема. Масштабованість за допомогою IVA обмежена лише обмеженням
уяви розробника.

4.2.3 Поліпшення інтелекту IVA.

В даний час ми використовуємо дуже базовий рівень коду для фільтрації
ключових слів, які розуміє IVA. Також виконуємо певні обмежені завдання
відповідно до голосових команд. IVA може стати ближчим до штучного нав-
чання та взаємодії на основі алгоритму машинного навчання. Оскільки машинне
навчання є найбільш актуальною темою. Воно забезпечує комп'ютеру здатність
фільтрувати візерунок, який надалі може бути використаний для виконання за-
вдань у певній галузі. Індустрія інформаційних технологій переживає бурхливі
дослідження в галузі машинного навчання. Ми також можемо побачити нові те-

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		54

хнології, що використовують їх для імітації обмеженого інтелекту, такого як ато-
пілот на автомобілях, цифровий помічник, медичні пристрої, що виявляють рак,
та багато інших. Ми можемо використовувати подібні підходи, щоб зробити ІВА
на крок ближче до розумної системи автоматизації.

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		55

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. AVC: How Big is Tablet Market: [Електронний ресурс] // Режим доступу до ресурсу: <http://avc.com/2014/04/how-big-is-the-tabletmarket/>
2. Home Automation history: [Електронний ресурс] // Режим доступу до ресурсу: http://en.wikipedia.org/wiki/Home_automation#History
3. Nest Thermostat: [Електронний ресурс] // Режим доступу до ресурсу: <https://nest.com/thermostat/meet-nest-thermostat>
4. Pew Research center: [Електронний ресурс] // Режим доступу до ресурсу: <http://www.pewInternet.org/2014/04/03/older-adults-andtechnology-use/>
5. Business College at University of Alabama: [Електронний ресурс] // Режим доступу до ресурсу: <http://businessdegrees.uab.edu/resources/infographics/the-future-of-mobile-application/>
6. Westinghouse Electric: [Електронний ресурс] // Режим доступу до ресурсу: http://en.wikipedia.org/wiki/Westinghouse_Electric_Company
7. Link-Labs: [Електронний ресурс] // Режим доступу до ресурсу: <http://www.link-labs.com/zigbee-vs-wifi-802-11ah/>
8. Axial Exchange on Older Americans use of smartphones: [Електронний ресурс] // Режим доступу до ресурсу: <http://axialexchange.com/blog/article/sanity-check-do-older-americans-use-smartphones>
9. Techcrunch article on smartphone market in developing markets: [Електронний ресурс] // Режим доступу до ресурсу: <http://techcrunch.com/2012/05/27/mobile-developing-world/>
10. Telegraph article on smartphone usage and future in developing countries: [Електронний ресурс] // Режим доступу до ресурсу:

<http://www.telegraph.co.uk/finance/newsbysector/mediatechnologyandtelecoms/11126997/Emerging-markets-will-lead-smartphone-growth-next-year.html>

11.Telecomengine: [Электронный ресурс] // Режим доступа до ресурсу:

<http://www.telecomengine.com/article/future-IoT-connectivity>

12.Arduino Wiki: [Электронный ресурс] // Режим доступа до ресурсу:

<https://arduino-info.wikispaces.com/MegaQuickRef>

13.Arduino - Home: [Электронный ресурс] // Режим доступа до ресурсу:

www.arduino.cc

14.HiLink Co, Ltd HLK-RM04 chip: [Электронный ресурс] // Режим доступа

до ресурсу: http://www.hlktech.net/product_detail.php?ProId=39

15.Google buys Nest news: [Электронный ресурс] // Режим доступа до ресурсу:

<http://www.independent.co.uk/life-style/gadgets-andtech/google-buys-nest-what-does-this-mean-for-home-automation-the-internet-ofthings-and-apple-9058217.html>

16.Android Developers: [Электронный ресурс] // Режим доступа до ресурсу:

<http://developer.android.com/index.html>

17.Programming questions and support: [Электронный ресурс] // Режим доступа до ресурсу:

<http://stackoverflow.com/>

18.Arduino and Android through Wi-Fi: [Электронный ресурс] // Режим доступа до ресурсу:

<http://electronics.stackexchange.com/questions/56633/simplest-way-to-connect-anrduino-to-android-wirelessly>

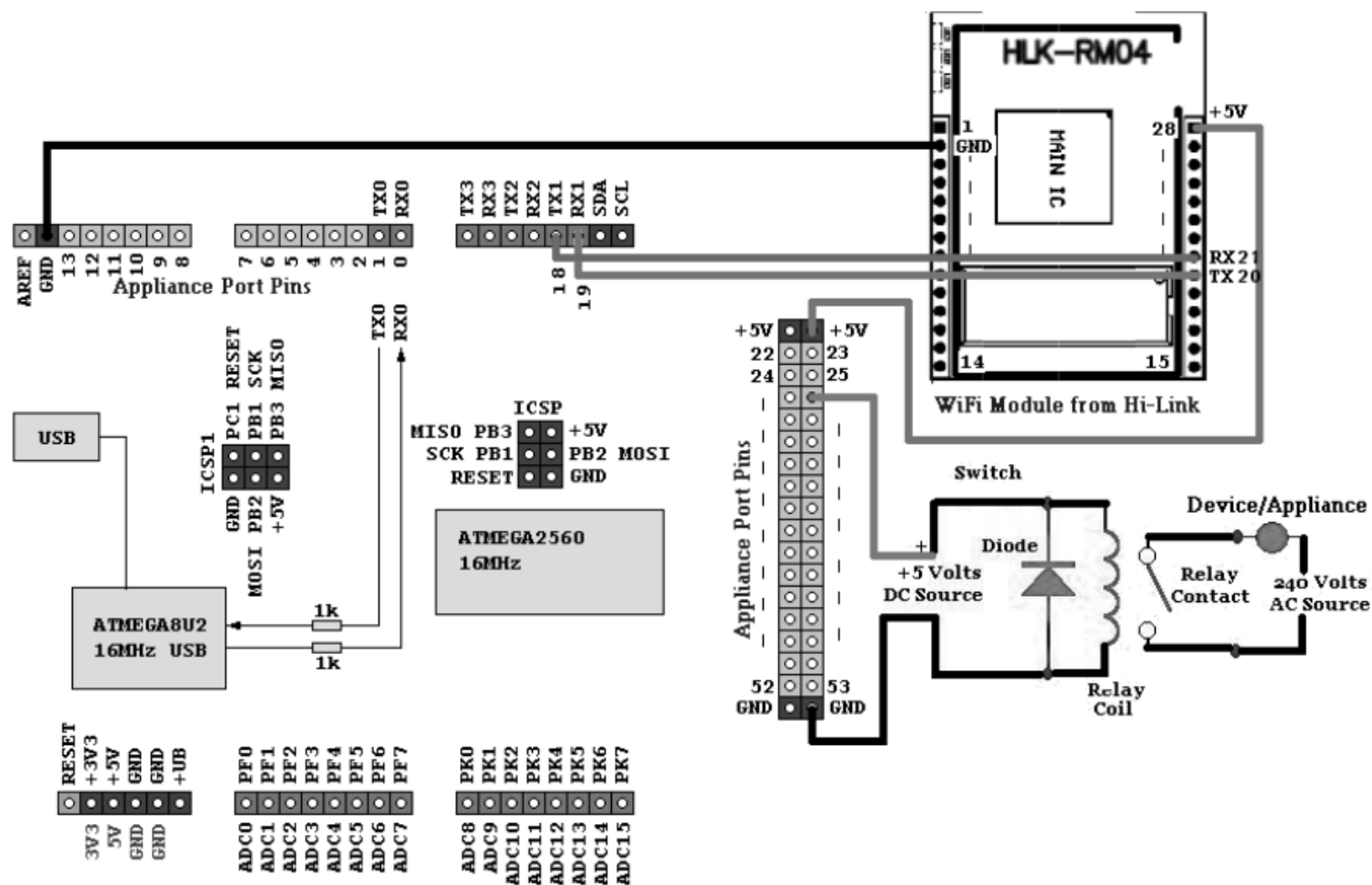
<http://electronics.stackexchange.com/questions/56633/simplest-way-to-connect-anrduino-to-android-wirelessly>

19.ABI Research on home automation future: [Электронный ресурс] // Режим доступа до ресурсу:

<https://www.abiresearch.com/press/15-million-home-automation-systems-installed-in-th>

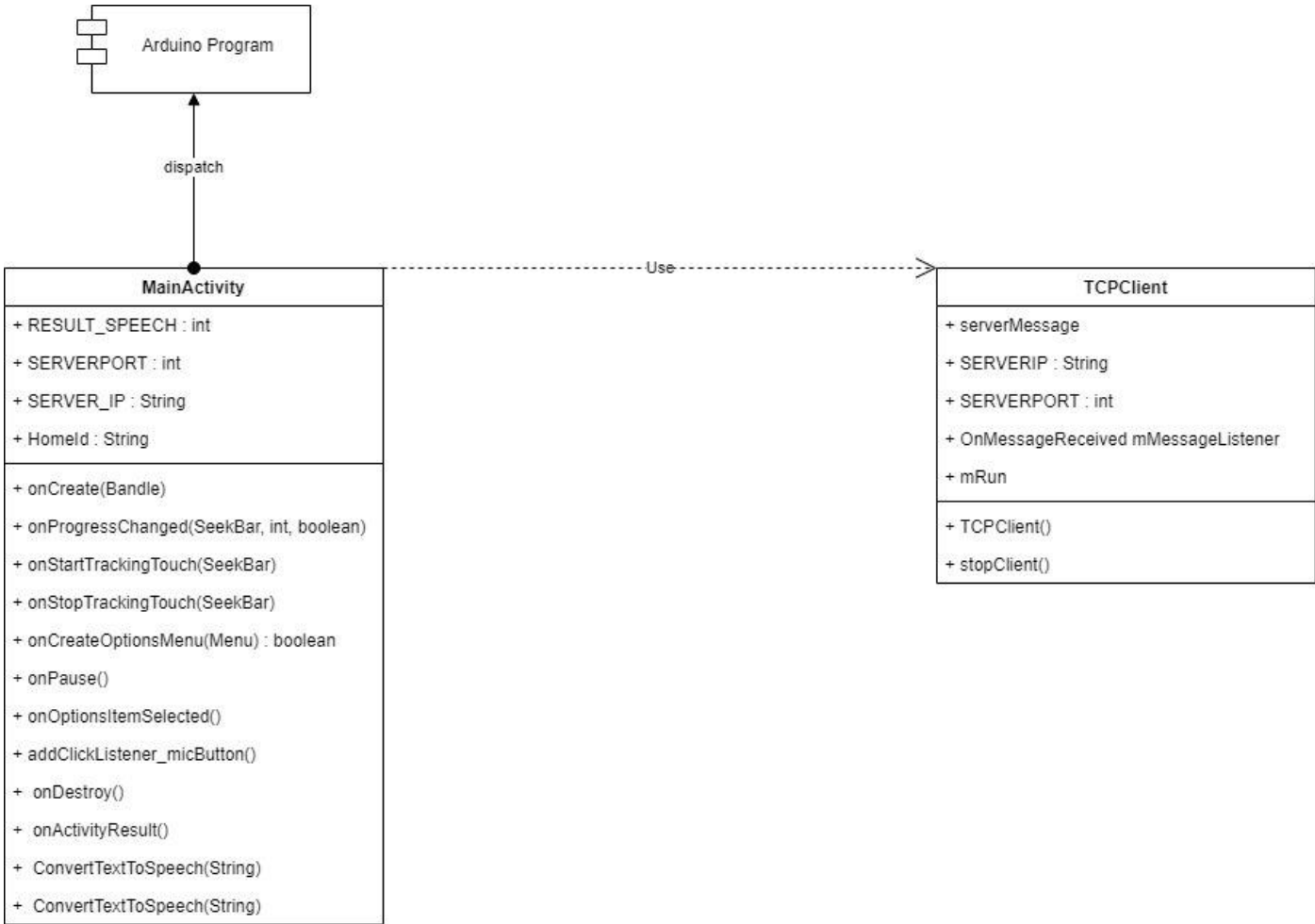
- 20.Speech Recognition by HowStuffWorks: [Електронний ресурс] // Режим доступу до ресурсу: <http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition1.htm>
- 21.Business Insider smartphone vs PC: [Електронний ресурс] // Режим доступу до ресурсу: <http://www.businessinsider.com/smartphonebigger-than-pc-market-2011-2>
- 22.Forbes: Mobile Is Eating the World: [Електронний ресурс] // Режим доступу до ресурсу:
<http://www.forbes.com/sites/louiscolumbus/2014/11/09/mobile-is-eating-the-world/>
- 23.Gigaom.com: [Електронний ресурс] // Режим доступу до ресурсу:
<https://gigaom.com/2014/10/03/belkin-looks-at-the-smart-home-anddoesnt-see-a-place-for-hubs/wikispace>
- 24.Arduino pinouts: [Електронний ресурс] // Режим доступу до ресурсу:
<https://arduino-info.wikispaces.com/MegaQuickRef>
- 25.ZigBee as Communication Platform for smart house applications: International Journal of Engineering - ISSN: 1584–2673

ДОДАТОК 1. Структурна схема пристрою



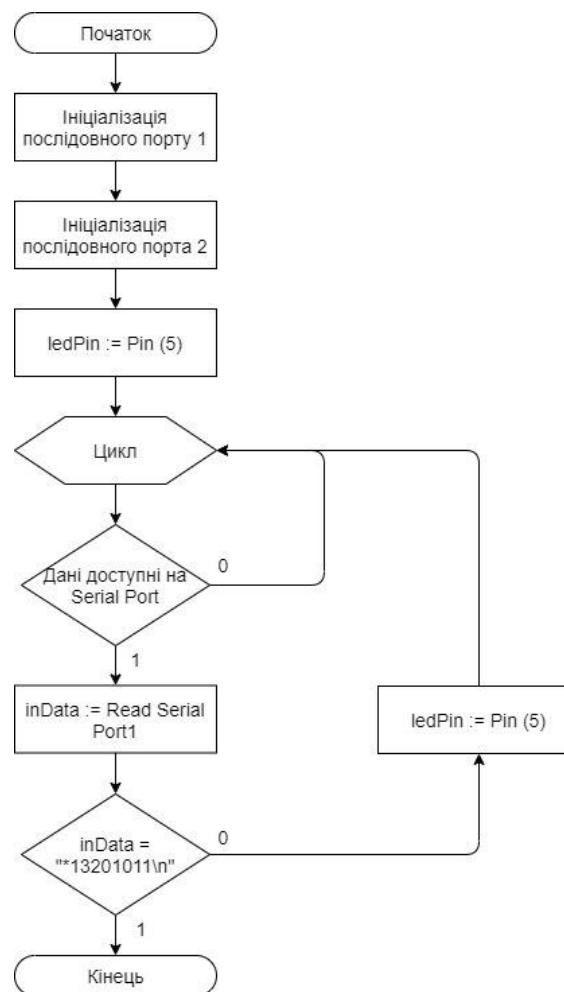
					ІАЛЦ466454.004.Д1						
					ДОДАТОК 1. Структурна схема пристрою	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив	Осадчий Ю. О.										
Перевірів	Русанова О. В.										
Т. контр											
						Лист 1			Листів 1		
Н.контр.	Сімоненко В. П.					ФІОТ Гр. ІО-64					
Затв.											

ДОДАТОК 2. Функціональна схема



					ІАЛЦ466454.005.Д2					
					ДОДАТОК 2. Функціональна схема	Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив	Осадчий Ю. О.									
Перевірів	Русанова О. В.									
Т. контр										
						Лист 1		Листів 1		
Н.контр.	Сімоненко В. П.					ФІОТ Гр. ІО-64				
Затв.										

ДОДАТОК 3. Принципова схема апаратного забезпечення



					ІАЛЦ466454.006.ДЗ								
					ДОДАТОК 3. Принципова схема апаратного забезпечення					Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата									
Розробив	Осадчий Ю. О.												
Перевірів	Русанова О. В.												
Т. контр										Лист 1		Листів 1	
										ФІОТ Гр. ІО-64			
Н.контр.	Сімоненко В. П.												
Затв.													

ДОДАТОК 4. ПРОГРАМНИЙ КОД ДОДАТКУ

Код програми Arduino.

```
void setup() {  
  int switch=27, index=0;  
  Serial1.begin(9600);  
  Serial2.begin(9600);  
  pinMode(ledPin, OUTPUT); // sets the digital pin as output for switching  
  // Pins 22, 24, 26 and 28 act as the controlling speed (regulating) the AC power  
  voltage //  
  regulation  
  pinMode(22, OUTPUT);  
  pinMode(24, OUTPUT);  
  pinMode(26, OUTPUT);  
  pinMode(28, OUTPUT);  
}  
void loop(){  
  if(Serial1.available()) {  
    int inByte = Serial1.read();  
    //Serial.write(inByte);  
    inData[index] = inByte; // Store it  
    index++; // Increment where to write next  
    inData[index] = '\0';  
    if(gotit==1)  
    {  
      index=0;  
      gotit=0;  
    }
```

					ІАЛЦ.467100.003 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	ДОДАТОК 4. Програмний код додатку	Літ.	Аркуш	Аркушів
Розробив		Осадчий Ю.О.						
Перевір.		Русанова О.В.					1	26
Н. контр.		Сімоненко В.П.						
Затверд.						НТУУ "КПР", ФІОТ, ІО-64		


```

}

if(inData[0]=='*' && inData[1]=='1' && inData[2]=='3' && inData[3]=='0'
&& in
Data[4]=='2' && inData[5]=='0' && inData[6]=='1' && inData[7]=='0' &&
inData[8]=='1' 15. && inData[9]=='1' && inData[10]=='\0' )
{
digitalWrite(switch, HIGH); // sets the Relay switch on
//Action performed. Now listen to new command
gotit=1;
index=0;
}
else if(inData[0]=='*' && inData[1]=='1' && inData[2]=='3' && in-
Data[3]=='0' &&
inData[4]=='2' && inData[5]=='0' && inData[6]=='1' && inData[7]=='0' &&
in
Data[8]=='1' && inData[9]=='0' && inData[10]=='\0' )//if(in-
Data=="*130201010\n")
{
digitalWrite(switch, LOW); // sets the Relay switch off
// Action performed. Now listen to new command
gotit=1;
index=0;
}
else if(inData[0]=='*' && inData[1]=='1' && inData[2]=='3' && in-
Data[3]=='0' &&

```

```

inData[4]=='2' && inData[5]=='0' && inData[6]=='1' && inData[7]=='0' &&
in
Data[8]=='2' && inData[9]=='0' && inData[10]=='\0' )//if(in-
Data=="*130201020\n")
{
Serial2.write(0); // write to dimmer module via serial port
gotit=1;
index=0;
}
else if(inData[0]=='*' && inData[1]=='1' && inData[2]=='3' && in-
Data[3]=='0' &&
inData[4]=='2' && inData[5]=='0' && inData[6]=='1' && inData[7]=='0' &&
inData[8]=='2' && inData[9]=='1' && inData[10]=='\0' )//if(in-
Data=="*130201021\n")
{
Serial2.write(45); // write to dimmer module via serial port
gotit=1;
index=0;
}
else if(inData[0]=='*' && inData[1]=='1' && inData[2]=='3' && in-
Data[3]=='0' &&
inData[4]=='2' && inData[5]=='0' && inData[6]=='1' && inData[7]=='0' &&
inData[8]=='2' && inData[9]=='2' && inData[10]=='\0' )//if(in-
Data=="*130201022\n")
{
Serial2.write(90); // write to dimmer module via serial port

```

```

gotit=1;
index=0;
}
else if(inData[0]=='*' && inData[1]=='1' && inData[2]=='3' && in-
Data[3]=='0' &&
inData[4]=='2' && inData[5]=='0' && inData[6]=='1' && inData[7]=='0' &&
inData[8]=='2' && inData[9]=='3' && inData[10]=='\0' )//if(in-
Data=="*130201023\n")
{
Serial2.write(135); // write to dimmer module via serial port
gotit=1;
index=0;
}
else if(inData[0]=='*' && inData[1]=='1' && inData[2]=='3' && in-
Data[3]=='0' &&
inData[4]=='2' && inData[5]=='0' && inData[6]=='1' && inData[7]=='0' &&
inData[8]=='2' && inData[9]=='4' && inData[10]=='\0' )//if(in-
Data=="*130201024\n")
{
Serial2.write(180); // write to dimmer module via serial port
gotit=1;
index=0;
}
else if(inData[0]=='*' && inData[1]=='1' && inData[2]=='3' && in-
Data[3]=='0' &&
inData[4]=='2' && inData[5]=='0' && inData[6]=='1' && inData[7]=='0' &&

```

```

inData[8]=='2' && inData[9]=='5' && inData[10]=='\0' )//if(in-
Data=="*130201025\n")
{
    Serial2.write(255); // write to dimmer module via serial port
    gotit=1;
    index=0;
}
}
}
}

```

Інтерфейс користувача для Android.

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/RelativeLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <ToggleButton
        android:id="@+id/toggleButton1"
        android:layout_width="102dp"
        android:layout_height="110dp"
        android:background="@drawable/light_off"
        android:text="LIGHTS(OFF)"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:textColor="#ff232323"
        android:longClickable="false"
        android:layout_alignParentTop="true" />
    <ToggleButton

```

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		5

```

android:id="@+id/toggleButton2"
android:layout_width="140dp"
android:layout_height="110dp"
android:background="@drawable/light_off"
android:text="COFFEE"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:textColor="#ff232323"
android:longClickable="false"
android:layout_alignParentTop="true" />

```

```

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:text="@string/voice_box"
android:textColor="#ffff2c25" />

```

```

<Button
android:id="@+id/button1"
android:layout_width="48dp"
android:layout_height="48dp"
android:background="@drawable/mic"
android:layout_above="@+id/textView1"
android:layout_centerHorizontal="true" />

```

```

<SeekBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/regulator1"
    android:layout_marginTop="36dp"
    android:layout_below="@+id/toggleButton1"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:max="5"
    android:layout_toStartOf="@+id/textView"
    style="@style/AppBaseTheme" />
</RelativeLayout>

```

Код Android java.

MainActivity.java

```

package com.example.androidsocketclient;

import android.app.Activity;
import android.content.ActivityNotFoundException;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.tts.TextToSpeech;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;

```

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		7

```

import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.ArrayList;
import java.util.Locale;
import android.os.Handler;
public class MainActivity extends Activity {
    protected static final int RESULT_SPEECH = 1;
    private static final int SERVERPORT = 8080;
    String SERVER_IP="10.0.0.14";
    String HomeId="*1302";
    public ToggleButton lamp1;
    public Button mic, button2;

```

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		8

```

public TextView voiceCommandBox, regulatorStatusLabel;
public EditText edText1;
public SeekBar regulator1;
public TextToSpeech tts;
String regulatorProgress="0";
Thread t;
Drawable lamp1_on_image, lamp1_off_image;
private Socket socket;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    SharedPreferences
shrPreferences=getApplicationContext().getSharedPreferences("UISettings",
Context.MODE_PRIVATE);
    SERVER_IP=shrPreferences.getString("IP_Address","");
    regulator1= (SeekBar) findViewById(R.id.regulator1);
    regulatorStatusLabel =(TextView) findViewById(R.id.textView);
    voiceCommandBox = (TextView) findViewById(R.id.textView1);
    mic = (Button) findViewById(R.id.button1);
    lamp1_on_image = getResources().getDrawable(R.drawable.light_on);
    lamp1_off_image = getResources().getDrawable(R.drawable.light_off);
    lamp1 = (ToggleButton) findViewById(R.id.toggleButton1);
    addClickListener_toggleButton();
    addClickListener_micButton();
    ConvertTextToSpeech("Hello there, I am IVA, smart home.");
    t = new Thread(new ClientThread());

```

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		9


```

t.start();
regulator1.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener()
{
    @Override
    public void onProgressChanged(SeekBar seekBar, int progresValue, boolean
b) {
        regulatorProgress = String.valueOf(progresValue);
        try {
            regulatorStatusLabel.setText(regulatorProgress);
            String command = HomeId + "0102" + regulatorProgress;
            PrintWriter out = new PrintWriter(new BufferedWriter(
            new OutputStreamWriter(socket.getOutputStream())),
            true);
            out.println(command);
        }
        catch (Exception e) {
            Toast.makeText(getApplicationContext(),"There is some problem connecting
to the device. Please check your connec-
tion",Toast.LENGTH_SHORT).show();
            e.printStackTrace();
            e.printStackTrace();
        }
    }
}

@Override
public void onStartTrackingTouch(SeekBar seekBar) {
}

```

```

@Override
public void onStopTrackingTouch(SeekBar seekBar) {
}
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
MenuInflater inflater = getMenuInflater();
inflater.inflate(R.menu.main, menu);
return true;
}

@Override
protected void onPause() {
if(tts != null){
tts.stop();
tts.shutdown();
}
super.onPause();
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
switch (item.getItemId()) {
case R.id.action_settings:
Intent refreshIntent = new Intent(getApplicationContext(),
Settings.class);
startActivity(refreshIntent);
break;

```

```

/*case R.id.action_refresh:
Intent refreshIntnt = new Intent(getApplicationContext(),
ThirdActivity.class);
startActivity(refreshIntnt);
break;*/
default:
break;
}
return super.onOptionsItemSelected(item);
}
public void addClickListener_micButton() {
mic.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent = new Intent(
RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "en-US");
try {
startActivityForResult(intent, RESULT_SPEECH);
voiceCommandBox.setText("");
} catch (ActivityNotFoundException a) {
Toast t = Toast.makeText(getApplicationContext(),
"Ops! Your device doesn't support Speech to Text",
Toast.LENGTH_SHORT);
t.show();
}
}
}
}

```

```

});
}

@Override
protected void onDestroy() {
    if(tts != null){
        tts.stop();
        tts.shutdown();
    }
    super.onDestroy();
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
        case RESULT_SPEECH: {
            if (resultCode == RESULT_OK && null != data) {
                ArrayList<String> text = data
                .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                voiceCommandBox.setText(text.get(0));
                filter(text.get(0).toString());
            }
            break;
        }
    }
}

private void ConvertTextToSpeech(final String text) {

```

```

// TODO Auto-generated method stub
/* if(text!=null||"".equals(text))
{
tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
}*/

tts=new TextToSpeech(MainActivity.this, new TextToSpeech.OnInitLis-
tener() {
@Override
public void onInit(int status) {
// TODO Auto-generated method stub
if(status == TextToSpeech.SUCCESS){
int result=tts.setLanguage(Locale.US);
if(result==TextToSpeech.LANG_MISSING_DATA ||
result==TextToSpeech.LANG_NOT_SUPPORTED){
Log.e("error", "This Language is not supported");
}
else{
//String text="hello there! I am Iva";
//ConvertTextToSpeech("hello there! I am Iva");
if(text!=null||"".equals(text))
{
tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
}
}
}
else
Log.e("error", "Initilization Failed!");
}
}
}

```

```

    }
    });
}

void filter(String line) {
    String sentence =line.toLowerCase();
    if( sentence.contains("dim") || sentence.contains("reduce") ||
sentence.contains("dimmer") || sentence.contains("less") || sentence.con-
tains("decrease")
|| sentence.contains("regulate")|| sentence.contains("regulator"))
    {
        if(sentence.contains("fan") || sentence.contains("light") ||
sentence.contains("lights") || sentence.contains("lamp") || sentence.con-
tains("intensity"))
        {
            int regProgress=Integer.parseInt(regulatorProgress);
            if(regProgress>0)
                regulator1.setProgress(regProgress-1);
            Toast to = Toast.makeText(getApplicationContext(),
"Regulator reduced!!!",
Toast.LENGTH_SHORT);
        }
    }
    else if(sentence.contains("increase") || sentence.contains("brighter") ||
sentence.contains("more") || sentence.contains("bright"))
    {
        if(sentence.contains("fan") || sentence.contains("light") ||

```

```

sentence.contains("lights") || sentence.contains("lamp") || sentence.contains("intensity"))
{
    int regProgress=Integer.parseInt(regulatorProgress);
    if(regProgress<5)
        regulator1.setProgress(regProgress+1);
    Toast to = Toast.makeText(getApplicationContext(),
        "Regulator Increased!!!",
        Toast.LENGTH_SHORT);
}
}

else if(sentence.contains("brew") || sentence.contains("caffeine") ||
sentence.contains("coffee") || sentence.contains("brewing") ||
sentence.contains("brewed"))
{
    if(!sentence.contains("not"))
        ConvertTextToSpeech("oh. . . something is wrong here, cannot find a coffee
        machine attached to the network.");
    else {
        ConvertTextToSpeech("Okay, I will not start brewing the coffee. Ohh! and by
        the way, I cannot find a coffee machine attached to my network.");
    }
}

else if (sentence.contains("lamp") || sentence.contains("light") ||
sentence.contains("bulb") || sentence.contains("lights") || (sentence.contains("dark"))) {

```

```

if (sentence.contains("so") || sentence.contains("too") || sentence.con-
tains("on")
&& !sentence.contains("not")) {
    boolean networkProblem=false;
    try {
        t = new Thread(new ClientThread());
        t.start();
        String str = HomeId+"01011";
        PrintWriter out = new PrintWriter(new BufferedWriter(
        new OutputStreamWriter(socket.getOutputStream())),
        true);
        out.println(str);
        socket.close();
        lamp1.setChecked(true);
        lamp1.setBackground(lamp1_on_image);
        ConvertTextToSpeech("The light is now on.");
        if(!networkProblem) {
            ConvertTextToSpeech("The light is now on.");
            Toast to = Toast.makeText(getApplicationContext(),
            "Light on!!!",
            Toast.LENGTH_SHORT);
            to.show();
        }
    } catch (Exception e) {
        networkProblem=true;
        ConvertTextToSpeech("There seems to be some problem connecting with the
network.");
    }
}

```



```

e.printStackTrace();
}
} else if ((sentence.contains("off") || sentence.contains("out") ||
sentence.contains("no") || (sentence.contains("dont") &&
sentence.contains("need"))||
(sentence.contains("no") && sentence.contains("need"))))
//&& !sentence.contains("not"))
{
try {
t = new Thread(new ClientThread());
t.start();
String str = HomeId+"01010";
PrintWriter out = new PrintWriter(new BufferedWriter(
new OutputStreamWriter(socket.getOutputStream())),
true);
out.println(str);
socket.close();
ConvertTextToSpeech("There you go. The light is off.");
Toast to = Toast.makeText(getApplicationContext(),
"Light off!!!",
Toast.LENGTH_SHORT);
to.show();
lamp1.setChecked(false);
lamp1.setBackground(lamp1_off_image);
} catch (Exception e) {
ConvertTextToSpeech("Cannot connect with the network. Check your LAN
connection.");
}
}

```

```

e.printStackTrace();
}
}
}
}

public void addClickListener_toggleButton() {
lamp1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
if (lamp1.isChecked()) {
try {
t = new Thread(new ClientThread());
t.start();

String str = HomeId+"01011";
// Thread readThread=new Thread(new ReadFromServerThread(socket));
// readThread.start();

PrintWriter out = new PrintWriter(new BufferedWriter(
new OutputStreamWriter(socket.getOutputStream())),
true);
out.println(str);

//
Toast.makeText(getApplicationContext-
text(),SERVER_IP,Toast.LENGTH_SHORT).show();
socket.close();

lamp1.setBackgroundDrawable(lamp1_on_image);
}catch (Exception e) {

```

```

lamp1.setChecked(false);
lamp1.setBackground(lamp1_off_image);
Toast.makeText(getApplicationContext(),"There is some problem
connecting to the device. Please check your
connection",Toast.LENGTH_SHORT).show();
e.printStackTrace();
}
} else {
try {
t = new Thread(new ClientThread());
t.start();
String str = HomeId+"01010";
// Thread readThread=new Thread(new ReadFromServerThread(socket));
// readThread.start();
PrintWriter out = new PrintWriter(new BufferedWriter(
new OutputStreamWriter(socket.getOutputStream())),
true);
out.println(str);
//
Toast.makeText(getApplicationContextCon-
text(),SERVER_IP,Toast.LENGTH_SHORT).show();
socket.close();
} catch (Exception e) {
lamp1.setChecked(false);
lamp1.setBackground(lamp1_off_image);
Toast.makeText(getApplicationContext(),"There is some problem
connecting to the device. Please check your

```

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		20

```

connection",Toast.LENGTH_SHORT).show();
e.printStackTrace();
}
lamp1.setBackground(lamp1_off_image);
}
}
});
}

class ClientThread implements Runnable {
@Override
public void run() {
try {
InetAddress serverAddr = InetAddress.getByName(SERVER_IP);
socket = new Socket(serverAddr, SERVERPORT);
} catch (UnknownHostException e1) {
e1.printStackTrace();
} catch (IOException e1) {
e1.printStackTrace();
}
}
}

class ReadFromServerThread implements Runnable {
private Socket clientSocket;
private BufferedReader input;
public ReadFromServerThread(Socket clientSocket) {
this.clientSocket = clientSocket;
try {

```

```
this.input = new BufferedReader(new
InputStreamReader(this.clientSocket.getInputStream()));
} catch (IOException e) {
e.printStackTrace();
}
}

public void run() {
while (!Thread.currentThread().isInterrupted()) {
try {
String read = input.readLine();
// updateConversationHandler.post(new updateUIThread(read));
Toast to = Toast.makeText(getApplicationContext(),
read,
Toast.LENGTH_SHORT);
to.show();
clientSocket.close();
} catch (IOException e) {
e.printStackTrace();
}
}
}
}
```

TCPClient.java

```
package com.example.androidsocketclient;
/**
 * Created by Mrunal
```

```

*/
import android.util.Log;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;
public class TCPClient {
    private String serverMessage;
    public String SERVERIP = "10.0.0.11"; //your computer IP address
    public int SERVERPORT = 8080;
    private OnMessageReceived mMessageListener = null;
    public boolean mRun = false;
    PrintWriter out;
    BufferedReader in;
    /**
     * Constructor of the class. OnMessageReceived listens for the messages re-
     ceived
     from server
     */
    public TCPClient(OnMessageReceived listener, String ip, int port) {
        mMessageListener = listener;
        mRun=false;
        if(ip!=null && !ip.equals(""))
        {

```

```

SERVERIP=ip;
}
if(port!=0)
{
SERVERPORT=port;
}
}
/**
 * Sends the message entered by client to the server
 * @param message text entered by client
 */
public void sendMessage(String message){
    Log.e("Client: Data->", message);
    if (out != null && !out.checkError()) {
        out.println(message);
        out.flush();
    }
}

public void stopClient(){
    mRun = false;
}

public void run() {
    mRun = true;
    try {
        //here you must put your computer's IP address.
        InetAddress serverAddr = InetAddress.getByName(SERVERIP);
        Log.e("TCP Client", "C: Connecting...");

```

```

//create a socket to make the connection with the server
Socket socket = new Socket(serverAddr, SERVERPORT);
try {
//send the message to the server
out = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream())), true);
Log.e("TCP Client", "C: Sent");
Log.e("TCP Client", "C: Done.");
//receive the message which the server sends back
in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
//in this while the client listens for the messages sent by the server
while (mRun) {
serverMessage = in.readLine();
if (serverMessage != null && mMessageListener != null) {
//call the method messageReceived from MyActivity class
mMessageListener.messageReceived(serverMessage);
Log.e("RESPONSE FROM SERVER", "S: Received Message: '" +
serverMessage + "'");
}
serverMessage = null;
}
} catch (Exception e) {
Log.e("TCP", "S: Error", e);
} finally {
//the socket must be closed. It is not possible to reconnect to this socket
// after it is closed, which means a new socket instance has to be created.
socket.close();

```



```

}
} catch (Exception e) {
Log.e("TCP", "C: Error", e);
}
}
}

```

//Declare the interface. The method messageReceived(String message) will
must be

implemented in the MyActivity

//class at on asynckTask doInBackground

```

public interface OnMessageReceived {
public void messageReceived(String message);
}
}

```

					ІАЛЦ.467100.003 ПЗ	Арк
Зм.	Арк.	№ докум.	Підпис	Дата		26